NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

INVISCID AERODYNAMIC PREDICTIONS OF HYPERSONIC ELLIPTICAL PROJECTILES: A COMPARATIVE STUDY OF THE EFFECTS OF STABILIZING SURFACES

by

Christopher Boyle

June 2000

Advisor: Beny Neta

Approved for public release; distribution is unlimited.

20000818 052

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, Va 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE June 2000	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Inviscid Aerodynamic Predictions of Hypersonic Elliptical Projectiles: A Comparative Study of the Effects of Stabilizing Surfaces	5. FUNDING NUMBERS
6. AUTHORS Boyle, Christopher	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT(*maximum 200 words*)

With the advent of "smart" munitions, the US and its allies are attempting to design more accurate tactical weapons. Of interest are relatively inexpensive barrel-launched projectiles capable of accuracy associated with guided munitions. This research studies potential configurations for a new class of kinetic energy projectiles. From past research, it has been shown that projectiles with elliptical cross-sections are more stable in flight than those with circular cross-sections. This research looks at one particular shape, an elliptical cone, and numerically predicts the aerodynamic attributes in inviscid, steady, hypersonic flow. In particular, the effects of different stabilizing surface configurations are evaluated. A residual benefit of this research is to show that ZEUS, an afterbody solver typically used for missile design, is capable of providing solutions for these configurations. The findings of this research will be delivered to the Chief, Aerodynamics Branch, Army Research Laboratory, and will fulfill part of ARL's commitment in a Key Technical Area agreement with the research laboratories of our allies.

14. SUBJECT TERMS CFD, Inviscid, Aerodynamics, Hypersonic, Projectiles, Strakes, Flares	15. NUMBER OF PAGES 168
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

i

THIS PAGE INTENTIONALLY LEFT BLANK

INVISCID AERODYNAMIC PREDICTIONS OF HYPERSONIC ELLIPTIC PROJECTILES: A COMPARATIVE STUDY OF THE EFFECTS OF STABILIZING SURFACES

Christopher Boyle
Captain, United States Army
B.S., United States Military Academy, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2000

Author: _____
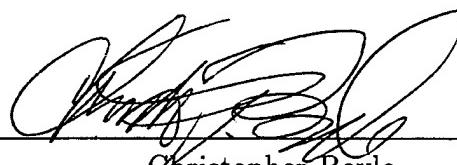
Christopher Boyle

Approved by: _____
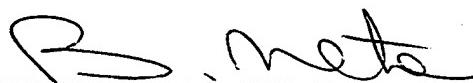
Beny Neta, Thesis Advisor

_____
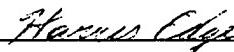
Harris Edge, Co-Advisor

_____
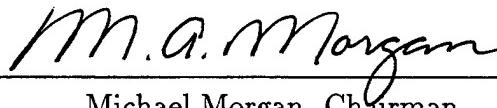
Michael Morgan, Chairman
Department of Mathematics

iii

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the advent of "smart" munitions, the US and its allies are attempting to design more accurate tactical weapons. Of interest are relatively inexpensive barrel-launched projectiles capable of accuracy associated with guided munitions. This research studies potential configurations for a new class of kinetic energy projectiles. From past research, it has been shown that projectiles with elliptical cross-sections are more stable in flight than those with circular cross-sections. This research looks at one particular shape, an elliptical cone, and numerically predicts the aerodynamic attributes in inviscid, steady, hypersonic flow. In particular, the effects of different stabilizing surface configurations are evaluated. A residual benefit of this research is to show that ZEUS, an afterbody solver typically used for missile design, is capable of providing solutions for these configurations. The findings of this research will be delivered to the Chief, Aerodynamics Branch, Army Research Laboratory, and will fulfill part of ARL's commitment in a Key Technical Area agreement with the research laboratories of our allies.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

Great care was taken to ensure the accuracy of the computer programs in the Appendix A, however they are supplied on an "as is" basis, with no warrantees of any kind. The author bears no responsibility for any consequences of using this program.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

ACKNOWLEDGMENTS

paper, technical report, dissertation, etc. that I requested. Despite my many requests for obscure publications, this staff never let me down.

I would be remiss if I failed to thank my office neighbor, Major Mike Huber. Both his professional and academic guidance have given me focus over the past two years. Throughout this project, his door has always been open to me and he has answered even the most mundane questions with enthusiasm and patience. I am looking forward to serving with him again in the future.

Finally, I would like to thank my primary advisor, Professor Beny Neta. His guidance throughout this project and during my two years at the Naval Postgraduate School has been instrumental to my intellectual growth. A patient yet demanding teacher, his high standards have driven me beyond my personal expectations. Without Professor Neta's uncompromising support, this work would have never come to fruition.

I.   INTRODUCTION

A.   BACKGROUND

In 1906, ten years prior to the first appearance of the tank at the Battle of Cambrai, an Austrian munitions manufacturer created the first armor piercing round for use in small caliber rifles and machine guns [Ref. 1]. At the time, the only combat systems employing armor were warships, so most of George Roth's peers thought him crazy. But early in World War I, ingenious soldiers, seeking protection from bombardment and waves of assaults, began using steel plates for protection. In an attempt to penetrate these plates, the Germans used Roth's patented rounds. In 1916, a new weapon, the tank, was employed on the battlefield and with its advent, munitions companies designed rounds that would defeat this new system. One of the first was created by Mauser. It was a 13mm rifle-fired projectile weighing 51.65g and had a muzzle velocity of 792 meters per second [Ref. 1].

With the passing of World War I, formal armor and mechanized units were created in the armies of the world's powers. The coming of age of the tank required an anti-tank weapon. In the 1920s and 1930s the general trend was to create weapons that fired larger projectiles or had greater muzzle velocity or both, exploiting the theory of kinetic energy. By the mid 1930s, the Germans and Poles each had weapons firing anti-tank projectiles in excess of 1,200 meters per second [Ref. 1].

Despite the heavy focus on high explosive (shape charge) munitions and the rocket launched projectile, the 1940s brought a significant breakthrough in armor piercing technology. The British formed a projectile using a dense tungsten core covered with a light alloy. The 57mm round was fired at 1,076 meters per second [Ref. 1]. Additionally, the British Armaments Research and Development Establishment recalled research performed by the French using discarding sabots on armor piercing rounds. The British exploited this research creating a 20mm, 1.47kg round capable of attaining a 1,234 meter per second muzzle velocity [Ref. 1]. This projectile type

1

became the primary armor piercing throughout the remainder of the war.

Following World War II, anti-armor technology was primarily focused on anti-tank guided missiles. It was not until the 1960s that a new development in armor piercing technology occurred. The Soviets believed that the rifling required to maintain aerodynamic stability usurped much of the energy of the propelling charge. Their solution was to use a smooth bore gun that propelled a fin stabilized dart. The dense sub-projectile was surrounded by sabots to fit the bore and capture the propellant gas behind it. This development reduced the bore friction allowing larger charges to be used and therefore achieve greater muzzle velocity.

By the 1970s, every mechanized army used the armor-piercing fin stabilized sabot (APFSDS), even those with rifled bores. The tungsten core was replaced by depleted uranium. Depleted uranium offered a similar density to tungsten carbide and was a by-product of the nuclear industry. Since this period, small modifications have been made to the kinetic energy round, however it is, in essence, the same projectile. The majority of improvements have been made to the delivery system, particularly the fire control system. By having an in depth understanding of the ballistic characteristics of the projectile, the fire control system can compensate for varying parameters (atmospheric conditions, gun tube wear, cant, etc).

Since the creation of the armor piercing round, many advancements in the shape, composition, and delivery have been made. However, one fact has remained unchanged; the kinetic energy round is an unguided, direct fire weapon. With "smart" munitions becoming increasingly common on the modern battlefield, the requirement for a new class of kinetic energy projectile has arisen. This new class of projectile may travel at very high speeds and require added maneuverability.

B.   STATEMENT OF THE PROBLEM

The motivation behind this research is to attempt to increase the accuracy of armor piercing projectiles with minimal additional munitions manufacturing cost

2

and with minimal modification to the launching platform. The main concept is to gain some of the accuracy associated with a "smart", guided missile with a relatively inexpensive "competent" kinetic energy round. This concept may be built upon in the future to allow for "smart" kinetic energy weapons.

Projectiles with unconventional shapes have been developed by British researchers. These projectiles are elliptic cones intended for travel at hypersonic speeds. In addition to the general elliptical shape, projectiles with added stabilizing surfaces have also been proposed, specifically, the addition of strakes and/or a tail flare. These shapes, designed for gun launched maneuver, are being analyzed by the United States, the United Kingdom, Canada, and Australia in accordance with a Key Technical Area (KTA) agreement. The United States Army Research Laboratory (ARL) is responsible for the computational fluid dynamics (CFD) analysis of several of these projectiles in the preliminary design stage.

In fulfillment of ARL's responsibility, this study will analyze four proposed projectile configurations in steady flow of an ideal gas. Computations will be performed at Mach 5, 6, 7, and 8.2 at angles of attack of 0°, 5°, 10°, and 15°. The goal is to predict the aerodynamic characteristics of each projectile and to provide a comparative analysis of the configurations. More specifically, the aerodynamic effects of the addition of stabilizing surfaces will be analyzed. The computational results will be compared to experimental results at Mach 8.2 and to computed data from previous research. Finally, this study will provide insight on the direction of future investigations into this class of weapon.

C.   PROJECTILE CONFIGURATIONS

The four projectiles of interest in this study are based on a standard configuration, a half power law elliptic cone measuring 540mm in length. This "baseline" configuration is free of stabilizing surfaces. It is designated the H4 (H3 in previous studies [Ref. 10]) and its dimensions are depicted in Figure 1. The second design
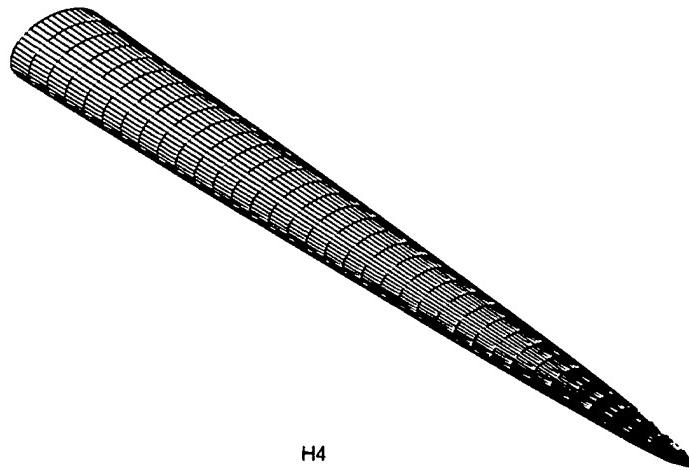
H4
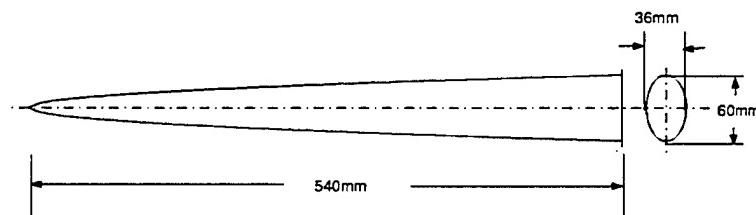
H4 from Atop
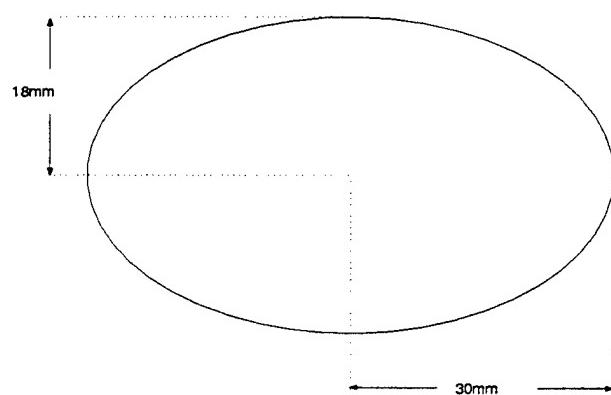
Cross-section at tail (z=540mm)

Figure 1. The H4 Projectile

adds four strakes to the H4 design. Each strake is tapered from the tail to zero at the nose following a constant circumferential angle. The strakes are symmetrically located on the body. This projectile is designated the H4A and is shown in Figure 2. The next configuration is identical to the H4 for the first 430mm in length. In the

H4A

Cross-section at tail (z=540mm)
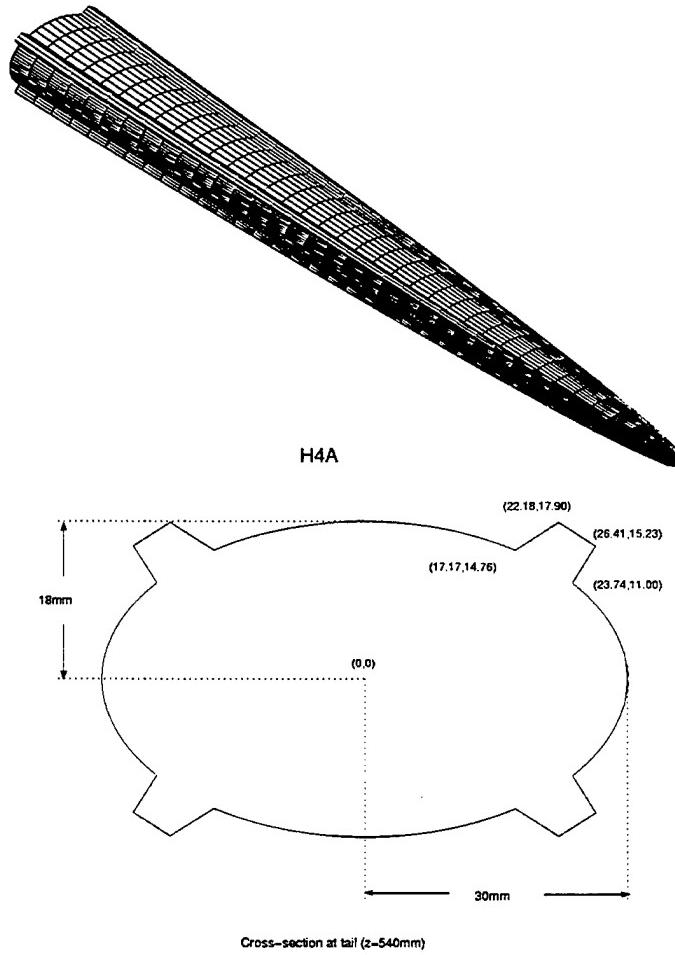
Figure 2. The H4A Projectile

110mm tail section, an elliptic flare is added. This flare is an elliptic conic section that increases the length of the semi-major axis at the base from 30mm as shown in Figure 1 to 35mm in Figure 3. The last configuration is a combination of the H4A and the H5, that is, a body with both strakes and a flare. The strake profile, identical to that of the H4A, blends into flare. Therefore, the strakes taper to zero (in height)

5

H5

45.7mm

70mm

430 mm   110 mm

H5 from Atop

22.85mm

35mm
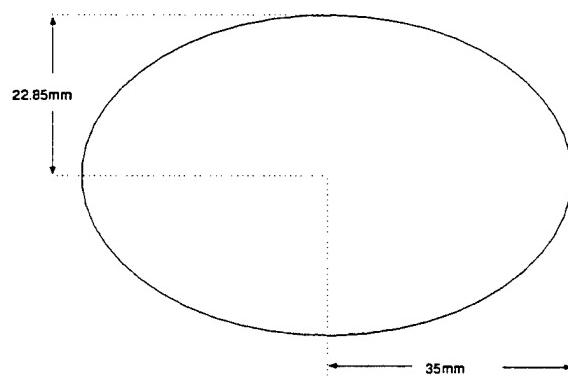
Cross-section at tail (z=540mm)

Figure 3. The H5 Projectile

at both the nose and tail of this projectile. This configuration, the H5A, is shown in Figure 4.

H5A
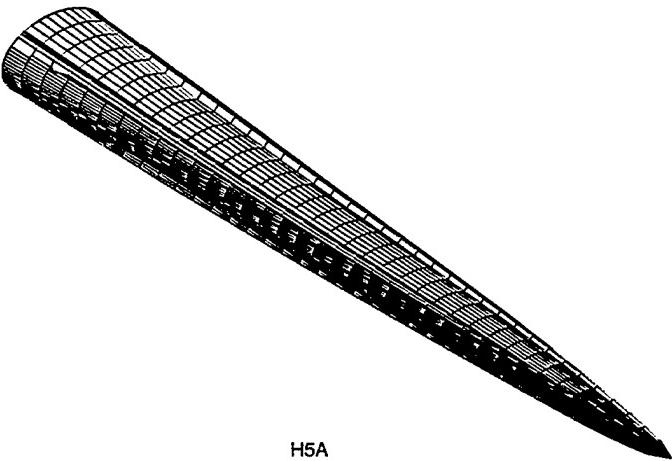
Figure 4. The H5A Projectile

D.   COMPUTATIONAL TOOLS

The primary solver used in this research is the Zonal Euler Solver (ZEUS) created by the Naval Surface Warfare Center for tactical missile design. This solver uses the inviscid Euler equations and a spatial marching technique to numerically approximate the flow field about the projectile. From the flow field characteristics, the aerodynamic forces and moments can be determined. In steady, supersonic flow, disturbances are only propagated downstream. Therefore, given an initial flow field on a cross field plane near the nose, the remaining flow field can be predicted by advancing (or marching) to a cross field plane further down the projectile axis. This type of solver requires supersonic conditions everywhere, a condition easily satisfied under the desired parameters of this study. The theory of this solution scheme is discussed in detail in Chapter II. This solver has a reputation as both an efficient and robust program [Ref. 2, 3, 4]. Implementation of this code is facilitated by the creation of the ZEUS Graphical User Interface (GUI) by Edge and Clarke at ARL [Ref. 5].

To obtain the initial flow field on a cross flow plane mentioned above, a nose solver was used. The blunt noses of the projectiles precluded the use of the nose solver included in the ZEUS package. However, a ZEUS-coupled version of the Conformal Mapping 3-Dimensional Transonic (CM3DT) code was used. This solver was originally designed for missile re-entry. It uses conformal mapping to produce a computational grid and solves the time-dependent, inviscid equations of motion in non-conservation form using a characteristic based finite difference scheme. By taking the time asymptotic limit, a steady solution is obtained and used as the nose boundary condition in ZEUS. Both CM3DT and ZEUS rely heavily on slender body theory and the assumptions of inviscid flow. In particular, at hypersonic speeds, the entire area of interest lies between the body and the bow shock wave. Slender body theory allows the solvers to establish boundaries based on these conditions. Additionally, the speeds at which the projectiles are being analyzed cause the boundary layer to become very thin, thereby, minimizing boundary layer effects.

To facilitate analysis of the results, researchers at ARL have written scripts for use with NASA's Flow Analysis Simulation Toolkit (FAST). These scripts enable the user to visualize the results obtained from both ZEUS and CM3DT. Additionally, prior to conducting computations, the scripts allow visualization of the computational grid, key in avoiding wasted computational time caused by poorly configured grids.

All computations were performed on Silicon Graphics Octane workstations at the Naval Postgraduate School in Monterey, California.

E.   PREVIOUS RESEARCH

The concept of non-circular munitions is far from novel. Research on bodies with elliptic cross-sections can be traced to the early 1950s. In Jack Nielsen's 1960 landmark work on missile aerodynamics, he dedicates a section to the aerodynamic theory of these shapes and discusses their aerodynamic advantages [Ref. 6].

Specific work began on power-law bodies in the 1960s and the advantages

8

of these shapes over circular cross-sections is well documented. Graves [Ref. 7] conducted a comparative study of the aerodynamic differences between a circular and elliptical (3:1 axes ratio) body. He concluded that the elliptical body generated approximately 25% more normal force than the circular body at supersonic speeds. This result was virtually independent of the angle of attack. Grave's study also showed that the elliptic shape had greater lateral and directional stability which was accentuated at higher angles of incidence.

Fournier et al. [Ref. 8] noted that increasing the ratio of the major axis to the minor axis increased lift. The research also showed that increasing Mach number (up to 4.63 in their research) had little effect on the $C_L - \alpha$ curve. Finally, they stated that the $C_L - \alpha$ curve was non-linear with the slope increasing with α. Kontis et al. [Ref. 9] maintained that this slope is further increased because of the separation of flow on the leeward side creating increased suction.

As the concept of a new class of kinetic energy projectile came to fruition, research on elliptic projectiles was revisited. Recent published research has been conducted primarily by British academics and researchers from the United Kingdom's Defense Research Agency (DRA). Of particular interest are the work by Edwards and Roper [Ref. 10] regarding the H4 and the works by Kontis et al. [Ref. 9, 11] comparing the H4 and H4A.

The work by Edwards and Roper analyzes the H4 computationally from Mach 5 to Mach 9 using a parabolized Navier-Stokes solver, CHASM, and the US Air Force's Missile DATCOM (theoretical techniques and semi-empirical approximations). The computational results were validated using experimental data obtained at Mach 8.2. This data will be used in this work to validate the accuracy of the solutions obtained by ZEUS.

The first work by Kontis et al. [Ref. 9] investigates the application of strakes to elliptic bodies. The H4 and the H4A are tested at Mach 8.2 at different angles of attack. In addition to the results of the gun tunnel experiment, schlieren pictures

were used to determine the location of the flow separation at different angles of attack. The results obtained by Kontis et al. [Ref. 9] are used here for comparison.

The other work by Kontis et al. [Ref. 11] analyzes a right elliptic cone with and without strakes in hypersonic flow. In this work, the authors conclude that in flow fields generated by these shapes, numerical studies can often produce valuable results despite being in poor agreement with experimental results. Their main contention being that although the numerical and experimental data did not agree, the computational data offers insights on the aerodynamic characteristics allowing better focus on further research topics.

At the time of the publication of this research, no experimental or computed data had been published on shapes similar to the H5 or H5A. However, numerous works have been conducted on hypersonic cone-cylinder-flare configurations. Of particular interest is the work by Hughson and Chapman [Ref. 12] comparing the hypersonic aerodynamics of two flare tail bodies, one with a cone nose and the other with a blunt spherical nose. This research found that the difference in bow shock waves caused by the nose shape leads to very different flow conditions at the flare.

Two additional studies pertaining to flare stabilization were applicable. The first by Guidos [Ref. 13] analyzes multiple cone-cylinder-flare configurations at Mach 4.5 to Mach 9.0 and at small angles of attack. Comparisons of the effects of flare length and angles are made. Noteworthy is the drag-stability trade-off. He notes that flared projectiles incur a severe drag penalty to maintain the stability levels of a fin stabilized projectile. However, at the desired speeds, the fins of the current kinetic energy rounds are expected to fail because of excessive loads and heating. The other work by Danberg et al. [Ref. 14] also compares multiple cone-cylinder-flare configurations including a flare with strakes (no strakes on the forebody). In this work particular attention is given to the effects of flare design on drag, stability, and force and pressure distributions.

The final work deserving note is that of Priolo and Wardlaw [Ref. 15], the co-

creators of ZEUS. In this study, three elliptic bodies were analyzed using experimental data and ZEUS. In this work, Priolo and Wardlaw describe techniques used in the application of ZEUS to bodies of elliptic cross-sections. The techniques used in Priolo and Wardlaw's work were used as a basis for the application of ZEUS in this study. Additionally, Priolo and Wardlaw document limitations of ZEUS when applied to non-circular cross-sections and describe the impact of these limitations on the results.

F.   SCOPE AND LIMITATIONS

Consistent with the design methodology proposed by Moore [Ref. 2], this study is a preliminary analysis of potential projectile configurations. This research is meant to provide a basis for further investigation or elimination of untenable designs. Because of the preliminary nature of this research, an efficient, solver was utilized allowing a blend of robustness, ease of use, and accuracy. The use of this solver incurred certain limitations. The solutions achieved are for steady, uniform flow conditions. Performance analysis in complex flow fields requires in depth study beyond the scope of this work. Additionally, the Euler equations (used in the solvers) cause an inherent neglect of the effects of the boundary layer (skin, friction, aero-heating, etc.). Investigation into these effects are left for future study. Finally, ideal gas conditions were used in this analysis, i.e. the effects of atmospheric conditions were neglected.

Because of time limitations, only the effects of changing angle of attack were analyzed. This research may be used as a basis for further study into the effects of roll or yaw changes.

Finally, because of the lack of experimental data for the H5 and H5A at the time of publication, the computed results are not validated. Based on the findings in this and other research reports, further experimental investigation may be conducted on these configurations.

THIS PAGE INTENTIONALLY LEFT BLANK

II.   MATHEMATICAL FOUNDATIONS OF THE SOLVERS

A.   CM3DT
1.   Background

The CM3DT code was created in 1979 by Darryl H. Hall [Ref. 16]. In its original form, the CM3DT code was designed to predict the flow field over missile nose tips during atmospheric re-entry (altitudes less than $50,000$ feet). It was found that during re-entry, the nose tips ablated. This ablation, coupled with the supersonic and hypersonic speeds at which the object traveled, required a solver that could treat blunting, its associated shock waves, and any potential embedded shock waves.

Under the conditions noted above (supersonic flow, low altitude, large Reynold's number), the shock is almost completely inviscid except for a thin boundary layer adjacent to the body surface. By assuming inviscid flow, the aerodynamic characteristics can be predicted by the numerical integration of the time dependent inviscid equations of motion. Additionally, at low altitudes, the inviscid pressure distribution provides an adequate method of predicting the forces and moments (except in the axial direction). It was also determined that this method could be used to solve for the steady flow field. By integrating the time dependent equations over time, an approximation to the steady solution can be found.

As inviscid Euler afterbody solvers were developed, the CM3DT code was implemented to solve the steady flow problem and provide steady initial conditions for the afterbody solvers. The code used in this research was modified specifically for use with the ZEUS code which does not have a built-in blunt nose solver.

CM3DT takes several steps to obtain a solution to the time dependent problem. First, CM3DT, through a series of conformal transformations in the projectiles' meridional planes, creates a new coordinate system that is closely aligned with the body surface. The coordinate system facilitates the prediction of the bow shock lo-

cation and velocity and the formulation of the mesh required for the application of a numerical scheme. This transformation is particularly important for non-simple or slender nose geometries such as those in this study. When this step is accomplished, a characteristic based scheme, the λ–differencing method [Ref. 20], is used to integrate the transformed, time-dependent equations of motion. This numerical technique applies a modified version of the MacCormack finite difference scheme and achieves second order accuracy.

When the time dependent equations are iteratively solved (in time), it is possible to achieve a reasonable representation of the steady flow solution. This code finds the solution for steady flow by taking an asymptotic limit of the unsteady flow solution.

This section describes the mathematical foundations used in CM3DT as applicable to this research. This description will omit ideas not pertinent to this particular problem. For a more detailed description, Hall [Ref. 16, 17] should be reviewed. The theory presented in this section is based on the research conducted by Moretti [Ref. 18, 20] and Hall [Ref. 17, 16, 19].

2.   Governing Equations in Cylindrical (x, y, ϕ) Space

The equations used in this code are the 3-D, time-dependent, inviscid equations of motion and based on the coordinate system shown in Figure 5. The equations are applied in the non-conservation form

$$P_t + UP_x + VP_y + \frac{WP_\phi}{y} + \gamma(V_x + V_y + \frac{W_\phi}{y} + \frac{V}{y}) = 0$$

$$U_t + UU_x + VU_y + \frac{WU_\phi}{y} + \frac{pP_x}{\rho} = 0$$

$$V_t + UV_x + \frac{WV_\phi}{y} - \frac{W^2}{y} + \frac{pP_y}{\rho} = 0$$

$$s_t + Us_x + Vs_y + \frac{Ws_\phi}{y} = 0$$

(II.1)

14
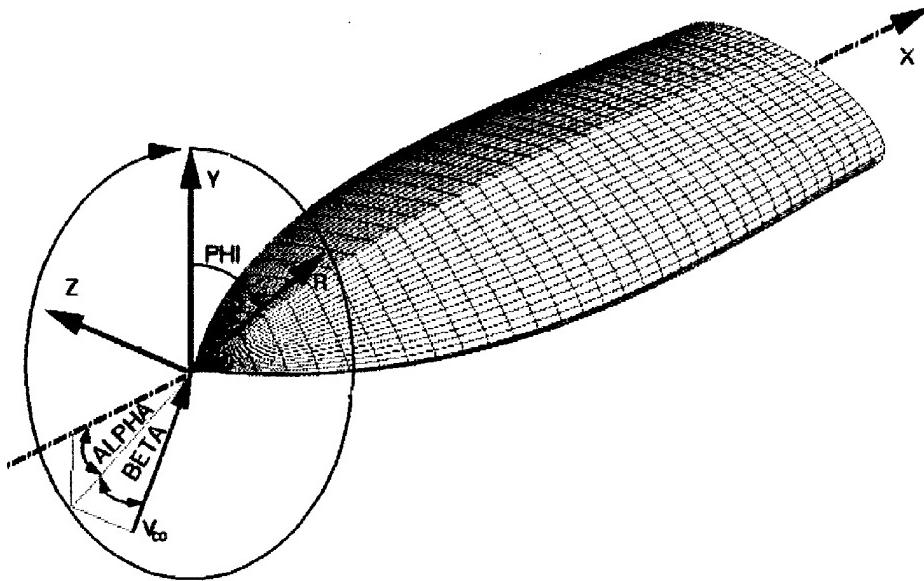
Figure 5. CM3DT Coordinate System

where P is the natural logarithm of pressure, p, γ is the isentropic exponent, s is entropy, and U, V, and W are velocity components in the x, y, and ϕ directions respectively. In these equations, the dependent variables are P, s, and the velocity components. The logarithm of pressure was chosen instead of pressure to minimize the large (orders of magnitude) changes in pressure. To close the system of equations, Hall uses the ideal form of the equation of state

$$\rho = p^{\frac{1}{\gamma}} e^{\frac{1-\gamma}{\gamma} s} \tag{II.2}$$

where

$$s = \frac{\bar{s} - \bar{s}_\infty}{R} \tag{II.3}$$

and R is the gas constant. To achieve a solution to the steady flow problem, boundary and initial conditions must be specified. Although any initial flow field will, in theory, converge to a unique steady state solution, to aid in convergence, Hall uses a technique

15

to provide a "reasonable" estimate. This technique is based on a user selected shock standoff distance which provides necessary shock information at the nose tip. In addition to the initial condition, two boundary conditions are required: the condition at the body surface; and the condition at the shock. The flow field on the body surface is determined numerically by applying the inviscid kinematic boundary condition (no velocity component normal to the body surface). The condition at the shock is found by numerically integrating the shock acceleration equation for the velocity and position of the wave. This information is unknown *a priori* and is determined during the numerical integration of the problem. It should be noted that no downstream boundary condition is required based on the assumption that the flow across this boundary is supersonic. According to slender body theory, there will be no upstream influence and, therefore, no requirement for a downstream boundary condition.

The equations and conditions described above are not applied in these forms. Hall uses a transformation technique developed by Moretti [Ref. 18] to create a more appropriate coordinate system. Consequently, these equations and conditions are mapped to a new system based on these conformal transformations.

3. Transformation of the Coordinate System and Construction of the Computational Grid

CM3DT applies a series of transformations such that the cylindrical coordinates (x, y, ϕ) are mapped to a new, near-cylindrical system (ξ, η, θ) that is closely aligned with the body surface of the projectile. In this transformation, ϕ is mapped directly to θ. Therefore, the true impact of this transformation occurs in the mapping of (x, y) to (ξ, η). In the transformed space, ξ represents the direction aligned with the surface of the body surface while η is normal to body surface. This orthogonality is guaranteed by the conformality of the mapping of the orthogonal space (x, y). It should be noted that this transformation is time-independent, so t is mapped directly

16

to τ, a time coordinate. The transformations can be represented generally as

$$\xi = f(x, y, \phi)$$
$$\eta = g(x, y, \phi)$$
$$\theta = \phi \qquad \qquad \text{(II.4)}$$
$$\tau = t$$

A representation of the untransformed and transformed coordinate systems (for any given ϕ) is shown in Figure 6a. To accomplish the transformation in each plane, Hall uses Moretti's "hinge point" concept [Ref. 18] extended to three dimensions. In this method, a finite number (I) of hinge points are generated such that the points approximate the meridional cross-section of the object. These points simulate the body contour in the (x, y) plane, that is, in each ϕ plane. CM3DT generates equally spaced hinge points based on the wetted length of the projectile. The hinge points are a uniform distance (user input) from the body surface. These hinge points create corners (see Figure 6a). First, using a series of transformations, each corner is sequentially smoothed (starting from the nose and working aft) to create a nearly horizontal line in the (ξ, η) plane. Although each transformation is applied to all hinge points, its effect is to map the next off-centerline hinge point to the horizontal axis without affecting the upstream (previously mapped) points on the horizontal axis. This process is depicted graphically in Figure 6 [Ref. 16]. Before continuing, the following notation is defined. Each hinge point is described as

$$h_{i,j} \qquad \qquad \text{(II.5)}$$

where the subscript denotes the ith hinge point and the jth transformation. Additionally, each space is defined in a complex plane such that the original space is defined as

$$z_1 = x + iy \qquad \qquad \text{(II.6)}$$

and

$$z_J = \hat{x} + i\hat{y} \qquad \qquad \text{(II.7)}$$
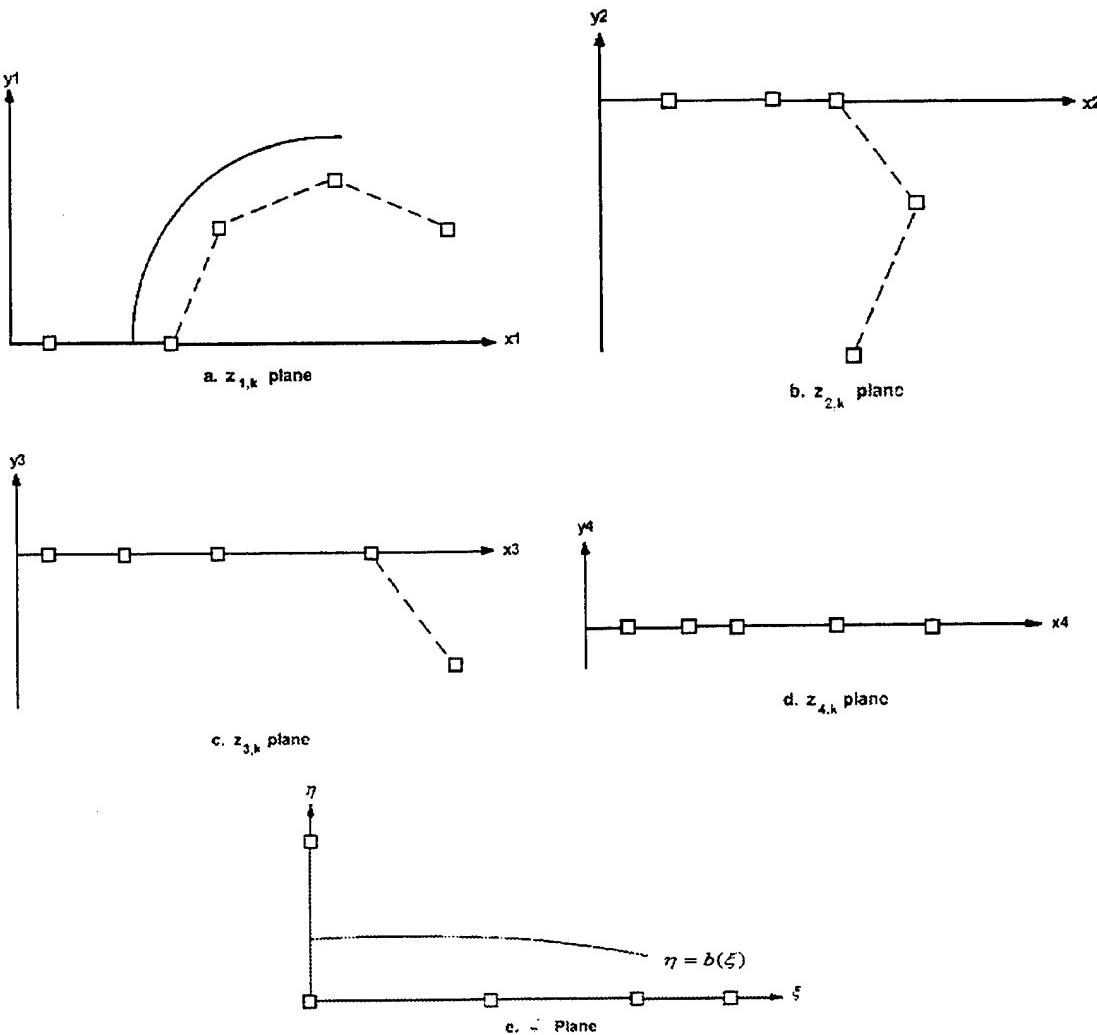
Figure 6. CM3DT Hinge Transformations

defines the space when the body surface is mapped to a horizontal line. J is the number of intermediate transformations required to map all hinge points to the horizontal axis. The CM3DT code requires that the first hinge point, $h_{1,j}$, and the second hinge point, $h_{2,j}$, both lie on the projectile centerline. The first point must lie outside of the nose surface and the second lies inside the surface (see Figure 6a). Since these first two points are on the horizontal axis, they do not require transformation. However, the remaining points will require a mapping. Therefore, the number of intermediate

18

mappings can be defined in terms of the number of hinge points

$$J = I - 2 \tag{II.8}$$

To accomplish the desired effect, the general form of each intermediate transformation is defined by

$$z_{j+1} = (z_j - h_{j+1,j})^{\delta_j} - 1 \tag{II.9}$$

where

$$\delta_j = \frac{\pi}{\pi - \arctan \dfrac{Im(h_{j+2,j} - h_{j+1,j})}{Re(h_{j+2,2} - h_{j+1,j})}} \tag{II.10}$$

Figure 6 depicts the complete transformation process. When all points are mapped to the horizontal axis, the space is z_J.

Now that the body surface has been mapped to the horizontal axis, it is useful to have this mapping perpendicular to the image of the centerline (the straight line between $h_{1,J}$ and $h_{2,J}$). Hall accomplishes this with a simple square root transformation

$$z_{J+1} = \sqrt{z_J - h_{2,J}} \tag{II.11}$$

One complication arising from the previous transformations must be resolved. Since the first two hinge points lie on the centerline, their location is independent of ϕ. The transformations thus far have been independent in each ϕ plane. Therefore, it is possible (and probable) that the images of the first two hinge points may be different for different ϕ. To ensure that all mappings of these points coincide, the following stretching is performed

$$\zeta = a_\phi z_{J+1} \tag{II.12}$$

where

$$a_\phi = \frac{h^1_{1,J+1}}{h^k_{1,J+1}} \tag{II.13}$$

and $h^k_{1,J+1}$ represents the first hinge point in the $J + 1$ space and in the kth ϕ plane. Having completed this series of conformal transformations, a new coordinate system

is defined where

$$\zeta = \xi + i\eta \tag{II.14}$$

and since ϕ was directly transformed to θ,

$$(x, y, \phi) \longmapsto (\xi, \eta, \theta) \tag{II.15}$$

is complete.

Knowing that the goal is to define a coordinate system that facilitates both the creation of a mesh and the application of a finite difference scheme, additional transformations are performed. Presently, the coordinate system (ξ, η, θ) is cylindrical-like. To create an equally spaced mesh, a rectangular system is required. Transformation to a Cartesian coordinate system also ensures easy numerical computation of derivatives. The transformation

$$
\begin{aligned}
X &= \frac{\theta}{2\pi} \\[2mm]
Y &= \frac{\xi}{\xi_L(\theta)} \\[2mm]
Z &= \frac{\eta - b(\xi, \theta)}{c(\xi, \theta, \tau) - b(\xi, \theta)} \\[2mm]
T &= \tau
\end{aligned} \tag{II.16}
$$

where $b(\xi, \theta)$ defines the body surface points, $c(\xi, \theta, \tau)$ defines the bow shock position, and ξ_L is the user defined downstream boundary. Body surface points are defined from a user input geometry file and the shock position is determined by numerical integration of the shock acceleration equation in time. This transformation has a normalizing effect on the coordinate system. That is, X becomes proportional to θ, Y varies between 0 (at the nose) and 1 (at the downstream boundary), and Z varies between 0 (on the body surface) and 1 (on the shock).

So far a series of intermediate conformal transformations have been performed. Since each of these mappings are conformal, the series of mappings results in a globally

conformal transformation. It should be noted that these transformations are time independent. Additionally, although the conformality of the global transformation guarantees that the orthogonal space (x, y) results in (ξ, η) being orthogonal, there is no assurance that (ξ, η, θ) is orthogonal.

Having defined the computational coordinate system, the mesh can be constructed. To build the grid, the user defines intervals in the X, Y, and Z directions. With these intervals established, the grid points are taken equally spaced in η, ξ, and θ. Figure 7 shows the grid generated for the H4/H5 nose at Mach 5 and $\alpha = 0^{\circ}$.
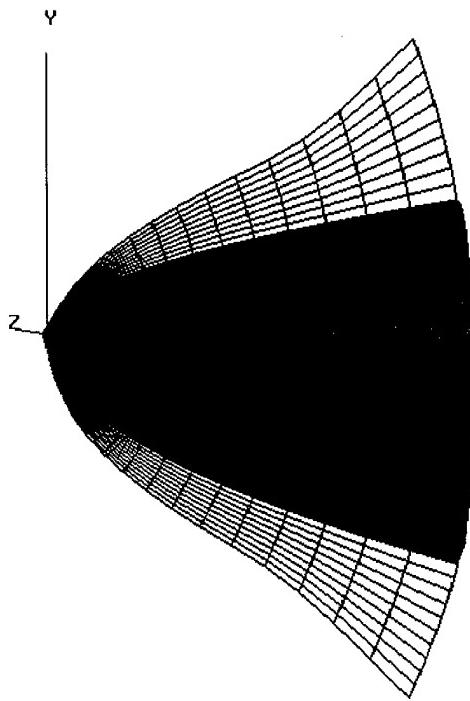
Figure 7. Example of a CM3DT Generated Grid (H4/H5 at $\alpha = 0^{\circ}$, Mach 5)

The change in the coordinate system affects the governing equations. In the next section, this effect is discussed.

4. Transformation of the Governing Equations

With a new coordinate system, it is necessary rewrite the governing equations. To convert from (x, y, ϕ) to (ξ, η, θ), the following chain rule relations are applied to (II.1)

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial \phi} = \xi_\phi \frac{\partial}{\partial \xi} + \eta_\phi \frac{\partial}{\partial \theta}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau}$$

$$\tag{II.17}$$

Now letting

$$g = \frac{\partial \zeta}{\partial z_1} = G e^{i\omega} = G \cos(-\omega) - iG \sin(-\omega) \tag{II.18}$$

and using the fact that the transformations are conformal the Cauchy-Riemann conditions are applied

$$\xi_x = \eta_y = Re(g)$$
$$\xi_y = -\eta_x = Im(g)$$

$$\tag{II.19}$$

Similarly, by defining

$$\phi = \frac{\partial \log g}{\partial \zeta} = \phi_1 + i\phi_2 \tag{II.20}$$

and transforming the velocity components

$$u = U \cos(\omega) - V \sin(\omega)$$
$$v = U \sin(\omega) + V \cos(\omega)$$
$$w = W$$

$$\tag{II.21}$$

22

the equations of motion and entropy can now be written as

$$\frac{DP}{D\tau} + \gamma G(u_\xi + v_\eta + v\phi_2 - u\phi_1)$$

$$+\gamma \frac{\xi_\phi w_\xi + \eta_\phi w_\eta + w_\theta - u\sin(\omega) + v\cos(\omega)}{y} = 0$$

$$\frac{Du}{D\tau} + \frac{vG}{y}(v\phi_1 + u\phi_2) + vw(\xi_\phi \phi_2 + \eta_\phi \phi_1 + \omega_\theta) + \frac{w^2 \sin(\omega)}{y}$$

$$+\frac{GpP_\xi}{\rho} = 0$$

$$\frac{Dv}{D\tau} + \frac{uG}{y}(v\phi_1 + u\phi_2) - uw(\xi_\phi \phi_2 + \eta_\phi \phi_1 + \omega_\theta) - \frac{w^2 \cos(\omega)}{y}$$

$$+\frac{GpP_\eta}{\rho} = 0$$

$$\frac{Dw}{D\tau} - \frac{w}{y}(u\sin(\omega) - v\cos(\omega)) + \frac{p}{\rho y}(\xi_\phi P_\xi + \eta_\phi P_\eta + P_\theta) = 0$$

$$\frac{Ds}{D\tau} = 0$$

(II.22)

where

$$\frac{D}{D\tau} = \frac{\partial}{\partial \tau} + (Gu + \frac{w\xi_\phi}{y})\frac{\partial}{\partial \xi} + (Gv + \frac{w\eta_\phi}{y})\frac{\partial}{\partial \eta} + \frac{w}{y}\frac{\partial}{\partial \theta} \qquad \text{(II.23)}$$

23

To transform (II.22) to the computational space (X, Y, Z), again the chain rule is applied

$$\frac{\partial}{\partial \tau} = \frac{\partial}{\partial T} + Z_\tau \frac{\partial}{\partial Z}$$

$$\frac{\partial}{\partial \xi} = Y_\xi \frac{\partial}{\partial Y} + Z_\xi \frac{\partial}{\partial Z}$$

(II.24)

$$\frac{\partial}{\partial \eta} = Z_\eta \frac{\partial}{\partial Z}$$

$$\frac{\partial}{\partial \theta} = X_\theta \frac{\partial}{\partial X} + Y_\theta \frac{\partial}{\partial Y} + Z_\theta \frac{\partial}{\partial Z}$$

where

$$X_\theta = \frac{1}{2\pi}$$

$$Y_\xi = \frac{1}{\xi_L(\theta)}$$

$$Y_\theta = -Y Y_\xi \xi_{L_\theta}$$

$$Z_\eta = \frac{1}{c - b}$$

(II.25)

$$Z_\xi = -Z_\eta [(1 - Z) b_\xi + Z c_\xi]$$

$$Z_\theta = -Z_\eta [(1 - Z) b_\theta + Z c_\theta]$$

$$Z_\tau = -Z Z_\eta c_\tau$$

and b and c are the body and shock positions and their derivatives represent the body and shock slopes. These relations allow the governing equations to be written in the

24

computational space

$$\frac{DP}{DT} + \gamma G[(Y_\xi u_Y + Z_\xi u_Z + Z_\eta v_Z + E - \frac{u\sin(\omega) - v\cos(\omega)}{Gy}] +$$

$$\gamma \frac{\xi_\phi(Y_\xi w_Y + Z_\xi w_Z) + \eta_\phi Z_\eta w_Z + X_\theta w_X + Y_\theta w_Y + Z_\theta w_Z}{y} = 0$$

$$\frac{Du}{DT} + vGD + \frac{vwF}{y} + \frac{w^2\sin(\omega)}{y} + \frac{Gp(Y_\xi P_Y + Z_\xi P_Z)}{\rho} = 0$$

$$\frac{Dv}{DT} - uGD - \frac{uwF}{y} - \frac{w^2\cos(\omega)}{y} + \frac{GpZ_\eta P_Z}{\rho} = 0 \qquad \text{(II.26)}$$

$$\frac{Dw}{DT} - \frac{w}{y}(u\sin(\omega) - v\cos(\omega))$$

$$+\frac{p}{\rho y}[(Z_\xi \xi_\phi + Z_\eta \eta_\phi + Z_\theta)P_Z + (\xi_\phi Y_\xi + Y_\theta)P_Y + X_\theta P_X] = 0$$

$$\frac{Ds}{DT} = 0$$

where

$$\frac{D}{DT} = \frac{\partial}{\partial T} + A\frac{\partial}{\partial Z} + B\frac{\partial}{\partial Y} + C\frac{\partial}{\partial X} \qquad \text{(II.27)}$$

and

$$A = Z_\tau + (Gu + \frac{w\xi_\phi}{y})Z_\xi + (Gv + \frac{w\eta_\phi}{y})Z_\eta + \frac{wZ_\theta}{y}$$

$$B = (Gu + \frac{w\xi_\phi}{y})Y_\xi + \frac{wY_\theta}{y}$$

$$C = \frac{wX_\theta}{y}$$

$$D = v\phi_1 + u\phi_2$$

$$E = v\phi_2 - u\phi_1$$

$$F = \xi_\phi\phi_2 + \eta_\phi\phi_1 + \omega_\theta$$

(II.28)

5. Applying the $\lambda-$Differencing Scheme

Having transformed the equations into a computational coordinate system, the $\lambda-$differencing scheme can be applied. This method is a characteristic based scheme for hyperbolic problems in which finite difference approximations of the derivatives are taken in the directions determined by the sign of the characteristic slope. By using this type of scheme, the method ensures that the wave propagation phenomena is more accurately modeled. In mathematical terms, the use of $\lambda-$differencing allows the domain of dependence information provided in the characteristic slopes to be incorporated into the finite difference scheme. To apply this method, the governing equations are combined with characteristic compatibility conditions. These new equations are integrated using a finite difference method and then the compatibility conditions are used to get the original form of the equations. Letting

$$
\begin{aligned}
\Lambda^+(\epsilon, \beta) &= \lambda_{\epsilon_1} \beta_{\epsilon_1} + \lambda_{\epsilon_2} \beta_{\epsilon_2} \\
\Lambda^-(\epsilon, \beta) &= \lambda_{\epsilon_2} \beta_{\epsilon_2} - \lambda_{\epsilon_1} \beta_{\epsilon_1}
\end{aligned}
\tag{II.29}
$$

the new form of (II.26) is

$$P_T + \tfrac{1}{2}\Lambda^+(Z,P) + \tfrac{1}{2}\Lambda^+(Y,P) + \tfrac{1}{2}\Lambda^+(X,P) + \tfrac{1}{2}\gamma\frac{\Lambda^-(Y,u)}{a\nu_Y} +$$

$$\tfrac{1}{2}\gamma\frac{\Lambda^-(Z,v)}{a\nu_Z} + \frac{1}{2}\gamma\frac{\Lambda^-(X,w)}{a} + \gamma G(Z_\xi u_Z + E + \frac{V}{Gy})$$

$$+\gamma\frac{(\xi_\phi Y_\xi + Y_\theta)w_Y + (\xi_\phi Z_\xi + \eta_\phi Z_\eta + Z_\theta)w_Z}{y} = 0$$

$$u_T + Au_Z + \tfrac{1}{2}\Lambda^+(Y,u) + Cu_X + vGD + \frac{vwF}{y}$$

$$+\frac{w^2\sin(\omega)}{y} + \frac{GpZ_\xi P_Z}{\rho} + \frac{1}{2}a\frac{\Lambda^-(Y,P)}{\gamma\nu_Y} = 0 \qquad\text{(II.30)}$$

$$v_T + \tfrac{1}{2}\Lambda^+(Z,v) + Bv_Y + Cv_X - uGD - \frac{uwF}{y}$$

$$-\frac{w^2\cos(\omega)}{y} + \frac{1}{2}a\frac{\Lambda^-(Z,P)}{\gamma\nu_Z} = 0$$

$$w_T + Aw_Z + Bw_Y + \tfrac{1}{2}(\Lambda^+(X,w) - w\frac{u\sin(\omega) - v\cos(\omega)}{y}$$

$$+p\frac{(\xi_\phi Z_\xi + Z_\eta\xi_\phi + Z_\theta)P_Z + (\xi_\phi Y_\xi + Y_\theta)P_Y}{\rho y} + \frac{1}{2}a\frac{\Lambda^-(X,P)}{\gamma} = 0$$

where

$$\nu_Z = \sqrt{1 + \left(\frac{Z_\xi}{Z_\eta}\right)^2 + \left(\frac{\xi_\phi\frac{Z_\xi}{Z_\eta} + \eta_\phi + \frac{Z_\theta}{Z_\eta}}{Gy}\right)^2}$$

$$\qquad\text{(II.31)}$$

$$\nu_Y = \sqrt{1 + \left(\frac{\xi_\phi + \frac{Y_\theta}{Y_\xi}}{Gy}\right)^2}$$

27

and the other coefficients are described by (II.28). λ is the slope of the characteristic curves in the $Z-T$, $Y-T$, and $X-T$ planes (subscripts denote the plane of reference). However, it is important to note that λ has two values (denoted by subscript), both of which are positive for supersonic flow. If the flow is subsonic, λ_1 is negative and λ_2 is positive. The sign of λ describes the direction of the domain of dependence at the point. For supersonic flow, only upwind information will be used in the finite difference derivative approximation of the spatial derivatives. However, for subsonic flow, both upwind and downwind information is used. A complete derivation of these equations and the compatibility conditions are described by Hall [Ref. 16] and Moretti [Ref. 20].

To integrate these equations, a modified version of the MacCormack method [Ref. 21] is used. This version of the MacCormack method is a one sided approach where in the predictor stage the backward and forward differences at $n\Delta x$ are respectively written

$$f_x = \frac{2f_n - 3f_{n-1} + f_{n-2}}{\Delta x}$$

$$f_x = \frac{f_{n+1} - f_n}{\Delta x} \qquad (II.32)$$

Similarly, the backward and forward corrected values are given by

$$\tilde{f}_x = \frac{\tilde{f}_n - \tilde{f}_{n-1}}{\Delta x}$$

$$\tilde{f}_x = \frac{\tilde{f}_{n+2} + 3\tilde{f}_{n+1} - 2\tilde{f}_n}{\Delta x} \qquad (II.33)$$

where $\tilde{f}$ denotes the predicted value. Because of the corrector step, this technique achieves second order accuracy. For supersonic flow, only upwind information is passed, so backward differences are used. For subsonic flow, the terms related to the λ_1 value use forward differences while the other terms use backward differences. The allowable time step size for this scheme is based on the Courant-Friedrichs-Lewy

28

(CFL) criterion. For stability, the step size is selected such that the numerical domain of dependence is contained in the physical domain of dependence.

6.   The Steady Solution

Using the methods from the previous section, the hyperbolic equations describing the time dependent solution are solved. For steady flow, it is assumed that over some given time, the time dependent solution will approach the steady solution. Therefore, by applying a time-asymptotic relaxation, Hall uses the time-dependent equations to achieve a solution for the steady flow problem, a more efficient method of obtaining the steady solution. The time dependent equations are hyperbolic and can be solved by forward marching (in time) as a mixed initial-boundary value problem. Conversely, the time independent equations are elliptic in nature and must be solved as a boundary value problem. As the time dependent equations are integrated in time, an adequate approximation of the steady solution is achieved. To determine when this iterative process reaches a sufficient representation, Hall establishes four convergence criteria. When these four criteria are met, the CM3DT code ends considering the final time iteration a close approximation to the steady solution. The criteria are:

1. The stagnation pressure is essentially constant. It must fall within a predesignated tolerance of the theoretical value.

2. The shock standoff distance must converge to a near constant value.

3. The root mean square of the shock velocities must be decreasing and the magnitude must be within a predesignated tolerance of the magnitude of the free stream velocity vector.

4. The total enthalpy at all points must be within a predesignated tolerance of the known steady state enthalpy (the steady state enthalpy is equal to the free stream total enthalpy).

When each of these conditions are met, the steady flow field is solved and can be used in an afterbody solver.

29

B. ZEUS

1. Background

ZEUS is an Euler solver created by the Naval Surface Warfare Center to perform aerodynamic predictions on tactical missiles with body-wing configurations. Since its implementation in 1986, ZEUS has been used extensively on both axisymmetric and non-axisymmetric configurations. The efficiency, robustness, and relative ease of use compared to Navier-Stokes solvers has made it popular with project managers particularly in preliminary design stages [Ref. 2]. For these reasons and its ability to handle thin, low aspect ratio lifting surfaces, the Army Research Laboratory has used it for aerodynamic predictions of projectiles.

ZEUS uses a second order Godunov finite volume scheme to explicitly integrate the inviscid Euler equations. Solutions are obtained by marching spatially from nose to tail. Like CM3DT, ZEUS generates a grid using the body surface and bow shock as boundaries. However, the grid accuracy is enhanced by the ability to divide the computational domain into one or more zones. Each zone can be mapped separately based on the cross plane geometry.

ZEUS requires several assumptions. First, ZEUS is a supersonic flow solver. The flow must be at least supersonic everywhere on the projectile such that disturbances do not affect upstream flow. Additionally, ZEUS assumes steady flow and marches spatially solving for a steady solution. Finally, since the inviscid Euler equations are used, the boundary layer is assumed to be very thin and its effects are neglected.

In this section, an overview of the methodology used by ZEUS will be presented. For an in-depth review of the theory, the reader should consult the work by Wardlaw et al. [Ref. 22, 23, 24].

2. Control Volume Notation and the ZEUS Coordinate System

Although mesh generation will be described later, it is important to establish the notation defining each control volume. In finite volume schemes, properties of cell edges (versus points) are calculated. Each cell is described by six edges. The size and location of each edge is defined by the location of a grid point which becomes the center of each cell face. The notation used in ZEUS for each point is (n, m, k) which is analogous to the ZEUS coordinate system (x, y, z). In ZEUS, x is up, y is left, and z points aft. Figure 8 [Ref. 24] shows the ZEUS coordinate system and the control volume notation. Having defined the cell notation, the Euler equations can
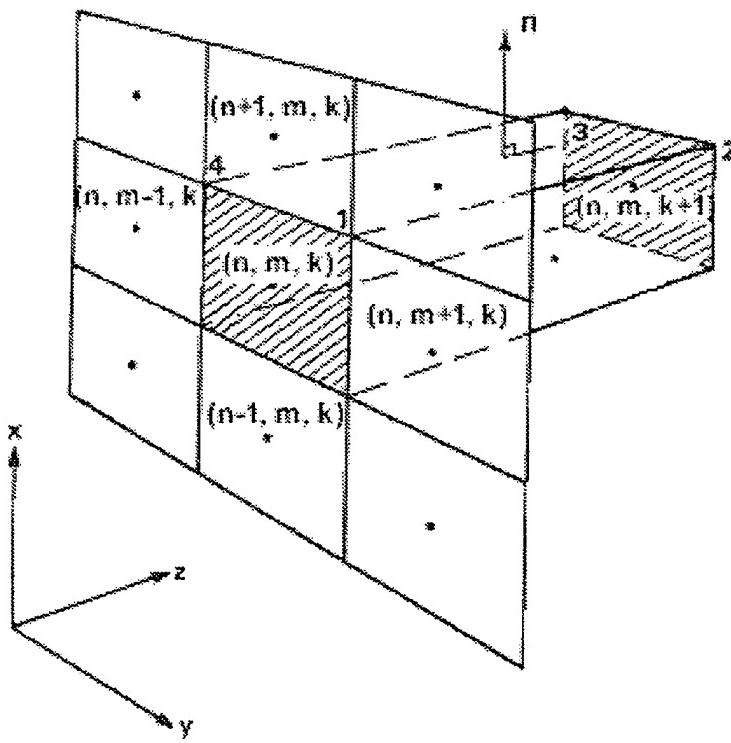
Figure 8. Control Volume Notation and ZEUS Coordinate System

31

be defined and converted to a usable form.

3.   Governing Equations

The inviscid steady flow Euler equations for perfect gases are defined as

$$\frac{\partial U}{\partial z} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \qquad \text{(II.34)}$$

where

$$U = \begin{pmatrix} \rho w \\ \rho w^2 + p \\ \rho w u \\ \rho w v \end{pmatrix}$$

$$F = \begin{pmatrix} \rho u \\ \rho u w \\ \rho u^2 + p \\ \rho u v \end{pmatrix} \qquad \text{(II.35)}$$

$$G = \begin{pmatrix} \rho v \\ \rho v w \\ \rho v u \\ \rho v^2 + p \end{pmatrix}.$$

These equations, based on the conservation of mass, momentum, and energy, can be placed in integral form

$$U_{n,m}^{k+1} - U_{n,m}^{k} + F_{n+\frac{1}{2},m} - F_{n-\frac{1}{2},m} + F_{n,m+\frac{1}{2}} - F_{n,m-\frac{1}{2}} = 0 \qquad \text{(II.36)}$$

where

$$U_{n,m}^k = A_{n,m}^k \begin{pmatrix} \rho w \\ \rho w^2 + p \\ \rho w u \\ \rho u v \end{pmatrix}_{n,m}^k = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

(II.37)

$$F_{n,m} = \begin{pmatrix} \rho V \\ \rho V w + p n_z \\ \rho V u + p n_x \\ \rho V v + p n_y \end{pmatrix}_{n,m}$$

In these equations, $A_{n,m}^k$ is the area of the cell edge containing (n, m, k), $\mathbf{n} = (n_x, n_y, n_z)$ is a vector normal to the cell edge (see Figure 8) and has a magnitude equal to the edge area, and $V = n_x u + n_y v + n_y w$. (u, v, w) are velocity vectors in the (x, y, z) directions respectively. U represents the flux passing through the shaded cell sides in Figure 8. That is, U is the flux passing through each z =constant cell face. Conversely, the F values represent the flux through the non-shaded sides. The systems found in (II.34) and (II.36) are in conservation form and are closed under the constraint based on the perfect gas equation of state and constant total enthalpy

$$H_0 = \frac{p}{\rho} \frac{\gamma}{\gamma - 1} + \frac{1}{2}(u^2 + v^2 + w^2)$$

(II.38)

Both equations (II.34) and (II.36) are valid for flows producing strong shocks.

4. Grid Generation

Having established the governing equations, it is now logical to define the computational grid. The grid generation in ZEUS differs from that of CM3DT in two ways:

- One simple transformation is used versus a series of conformal transformations
- Grid points serve as defining points for control volumes

ZEUS avoids the need for conformal mappings by allowing the user to divide each cross flow plane into quadrilateral zones thereby providing a method for dealing with

33

complex geometries. Typically, this zoning is used for wing-body configurations. In this work, all grids consist of a single zone. The zone boundaries in these cases are the body surfaces, the bow shocks, and the leeside and windside meridians. Figure 9 [Ref. 24] defines the ZEUS notation for each zone.
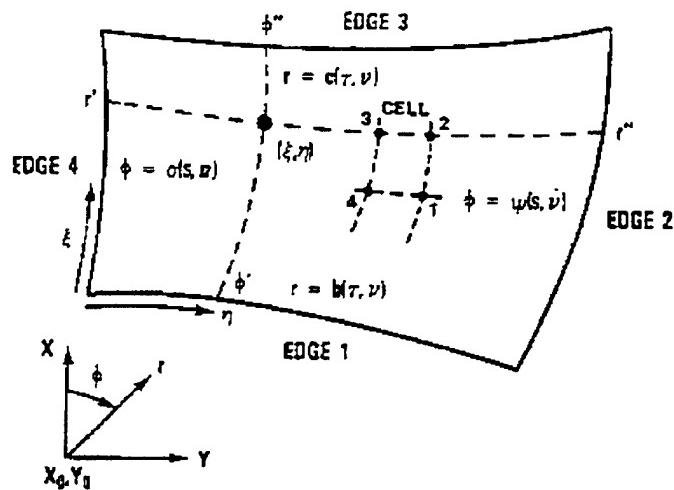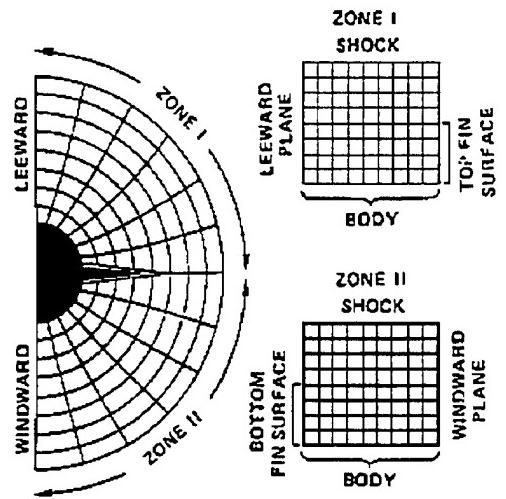
Figure 9. ZEUS Zone and Mesh Notation

ZEUS creates the mesh using the mapping $T_1 \cdot T_2$ where

$$T_1: \quad (\xi, \eta, \zeta) \mapsto (s, \tau, \nu)$$
$$T_2: \quad (s, \tau, \nu) \mapsto (x, y, z) \tag{II.39}$$

In Cartesian coordinates, this transformation becomes

$$x = s$$
$$y = \tau \tag{II.40}$$
$$z = \nu$$

and

$$x = b(\tau', \zeta) + [c(\tau'', \zeta) - b(\tau', \zeta)] f(\xi)$$
$$y = \sigma(s', \zeta) + [\psi(s'', \zeta) - \sigma(s', \zeta)] g(\eta) \tag{II.41}$$

where the points σ, b, c, and ψ are shown in Figure 9 and are described by

$$\tau' = \tau_4(\zeta) + [\tau_1(\zeta) - \tau_4(\zeta)] g(\eta)$$
$$\tau'' = \tau_3(\zeta) + [\tau_2(\zeta) - \tau_3(\zeta)] g(\eta)$$
$$s' = s_4(\zeta) + [s_3(\zeta) - s_4(\zeta)] f(\xi)$$
$$s'' = s_1(\zeta) + [s_2(\zeta) - s_1(\zeta)] f(\xi) \tag{II.42}$$

and (s_i, τ_i) are coordinates of the cell corners. The functions $f(\xi)$ and $g(\eta)$ determine the clustering of the mesh. The user defines these functions by designating arc length percentages at each zone edge. For a uniform mesh,

$$f(\xi) = \xi$$
$$g(\eta) = \eta \tag{II.43}$$

In this study, the grid was clustered only in the circumferential direction.

5. Transforming the Governing Equations

Having established a computational grid based on the (ξ, η, ζ) coordinate system, a final transformation of the governing equations can be performed.

35

In order to evaluate the terms in (II.36), this system must be put in non-conservation form

$$u_\zeta = \frac{-1}{w}\left(u_\xi \tilde{U} + u_\eta \tilde{V} + \frac{\xi_x p_\xi + \eta_x p_\eta}{\rho}\right)$$

$$v_\zeta = \frac{-1}{w}\left(v_\xi \tilde{U} + v_\eta \tilde{V} + \frac{\xi_y p_\xi + \eta_y p_\eta}{\rho}\right)$$

$$\text{(II.44)}$$

$$w_\zeta = \frac{1}{w^2 - a^2}(H_1 - wH_2 + \frac{P}{\rho})$$

$$p_\zeta = \frac{1}{w^2 - a^2}(-\rho w H_1 + \rho a^2 H_2 - wP) - \xi_z p_\xi - \eta_z p_\eta$$

where

$$\tilde{U} = \xi_x u + \xi_y v + \xi_z w$$

$$\tilde{V} = \eta_x u + \eta_y v + \eta_z w$$

$$P = u(\xi_x p_\xi + \eta_x p_\eta) + v(\xi_y p_\xi + \eta_y p_\eta) \qquad \text{(II.45)}$$

$$H_1 = a^2 \left(\xi_x u_\xi + \xi_y v_\xi + \xi_z w_\xi + \eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta\right)$$

$$H_2 = \tilde{U} w_\xi + \tilde{V} w_\eta$$

It should be noted that Z is mapped directly to ζ. Derivation of this form of the Euler equations is performed by using a non-linear transformation of variables. The variables in this system can be evaluated using the numerical scheme described in the following section.

6. The Numerical Scheme: An Application of a Second Order Godunov Method

To numerically solve the non-linear systems,(II.36) and (II.44), a second order Godunov predictor-corrector scheme is used. This method determines the fluxes, F, given in (II.36). This is accomplished by evaluating (II.44) using a twelve step method described by Wardlaw and Davis [Ref. 22]. For a more detailed (and theoretical) description of the method, Davis [Ref. 28] should be reviewed.

36

The first step in this process computes the partial derivatives of the primitive variables, (p, ρ, u, v, and w), with respect to the computational coordinates, (ξ and η). These values are obtained using a limiter and by computing the central differences. The role of the limiter will be discussed in Chapter V. However, it should be noted that the value of the limiter is usually one for all interior points. Letting f represent any of the primitive variables and φ represent the computational coordinates, an approximation of the partial derivatives is obtained using the equation

$$\frac{\partial f}{\partial \varphi}_{n,m} = \begin{cases} 0 & \Delta f_{n,m} \nabla f_{n,m} < 0 \\ \frac{M}{\Delta \varphi} \min \left[\frac{|\nabla f_{n,m} + \Delta f_{n,m}|}{2}, K|\Delta f_{n,m}|, K|\nabla f_{n,m}| \right] & \text{otherwise} \end{cases}$$

(II.46)

where

$$\begin{aligned}
\Delta f_{n,m} &= f_{n+1,m} - f_{n,m} \\
\nabla f_{n,m} &= f_{n,m} - f_{n-1,m} \\
M &= \text{sgn}(\nabla f_{n,m})
\end{aligned}$$

(II.47)

and n and m represent the nth and m^{th} points in the η and ξ directions respectively. This computation is performed for each point in the $z =$ constant plane.

Similarly, the metrics, ξ_x, ξ_y, ξ_z, η_x, η_y, η_z, ζ_x, ζ_y, and ζ_z are evaluated at each point. These values are used in the predictor step and in determining the step size, Δz. The values of the metrics are obtained analytically using the equations

$$\xi_x = \frac{\tau_\eta}{j} \qquad \xi_y = \frac{-s_\eta}{j} \qquad \xi_z = \frac{s_\eta \tau_\zeta - s_\zeta \tau_\eta}{j}$$

$$\eta_x = \frac{-\tau_\xi}{j} \qquad \eta_y = \frac{s_\xi}{j} \qquad \eta_z = \frac{s_\zeta \tau_\xi - s_\xi \tau_\zeta}{j}$$

(II.48)

$$\zeta_x = 0 \qquad \zeta_y = 0 \qquad \zeta_z = 1$$

where

$$j = s_\xi \tau_\eta - s_\eta \tau_\xi$$

(II.49)

37

and

$$s_\xi = (c - b)f'(\xi)$$

$$s_\eta = b_\tau \tau'_\eta f(\xi) + c_\tau \tau''_\eta f(\xi)$$

$$s_\zeta = (b_\tau \tau'_\zeta + b_\nu)(1 - f(\xi)) + (c_\tau \tau''_\zeta + c_\nu)f(\xi)$$

$$\tau_\xi = \sigma_\tau s'_\zeta(1 - g(\eta)) + \psi_s s'_\zeta g(\eta) \tag{II.50}$$

$$\tau_\eta = (\psi - \sigma)g'(\eta)$$

$$\tau_\zeta = (\sigma_s s'_\tau + \sigma_\nu)(1 - g(\eta)) + (\psi_s s''_\tau + \psi_\nu)g(\eta)$$

Details of this derivation are in Wardlaw and Davis [Ref. 22]. Having computed the partials of the primitive variables and the metrics, the step size can be calculated.

The step size, Δz, is determined by the CFL condition.

$$\Delta z = \frac{(w^2 - a^2)\Delta \eta}{\delta c_1 + c_2 + \sqrt{a^2(c_3 \delta^2 + c_4 + c_5 \delta)}} \tag{II.51}$$

where

$$c_1 = |a^2 \xi_z - w\tilde{U}|$$

$$c_2 = |a^2 \eta_z - w\tilde{V}|$$

$$c_3 = (w\xi_z - \tilde{U})^2 + (w^2 - a^2)(\xi_x^2 + \xi_y^2)$$

$$c_4 = (w\eta_z - \tilde{V})^2 + (w^2 - a^2)(\eta_x^2 + \eta_y^2) \tag{II.52}$$

$$c_5 = 2|(w\xi_z - \tilde{U})(w\eta_z - \tilde{V}) + (w^2 - a^2)(\eta_x \xi_x + \eta_y \xi_y)|$$

$$\delta = \frac{\Delta \eta}{\Delta \xi}$$

and a is the speed of sound. This condition ensures that the domain of dependence of the numerical scheme contains the domain of dependence of the Euler equations. ZEUS allows for user modification of the step size using a CFL safety factor. Application of this factor will be discussed later.

At this point, the step size, the metrics, and the primitive variable partial derivatives have been calculated. With this information, the solution can be advanced and the predictor values of the primitive variables can be calculated. The calculation of the predictor values is taken at $z + \frac{\Delta z}{2}$. The predictor values of the primitive variables, $\bar{p}, \bar{\rho}, \bar{u}, \bar{v}$, and $\bar{w}$, are calculated by substituting the metrics and derivatives into (II.44) and solving the system.

38

Before solving for the corrector values, information describing the control volumes and cells must be computed. The goal of this step is to calculate the coordinates of the control volume corners and to determine the vectors normal to the cell edges (see Figure 8). To determine the coordinates for the corners of the control volume, the grid generation transformation is used (II.41). The normal vectors for each cell edge are calculated using

$$\mathbf{n}_{n+\frac{1}{2},m} = \frac{1}{2}(\mathbf{V_{24}} \times \mathbf{V_{31}})$$ (II.53)

where $\mathbf{V_{n,m}}$ is the vector from (x_n, y_n, z_n) to (x_m, y_m, z_m) and the subscripts correlate to the corners of each cell edge in Figure 8.

Two critical pieces of the problem are known: the predictor values of the primitive variables and the location of the cell corners. From these pieces of information, the predictor values at the cell edges can be interpolated. However, each interior edge is influenced by two points. For example, in Figure 8, the lower edge of the cell defined by (n, m, k) is influence by both (n, m, k) and $(n-1, m, k)$. This will be resolved in the next step. To interpolate the values at the cell edges (using the notation in II.46), the following equations are used

$$f_{n,m} - \frac{1}{2}\frac{\partial f}{\partial \varphi}_{n,m}$$

(II.54)

$$f_{n-1,m} + \frac{1}{2}\frac{\partial f}{\partial \varphi}_{n-1,m}$$

To determine, the actual value of each property at the cell edges, the Riemann problem is constructed. For supersonic flow, the Riemann problem represents the confluence of two supersonic streams. At the intersection of these streams, both are turned to a common direction and have identical pressures. Figure 10 depicts pressure-direction curves for the confluence of two streams. Solving the Riemann problem results in constant properties for both streams. By using this technique and solving the Riemann problem, the properties at the cell edges can be found. ZEUS has the ability to solve the complete Riemann Problem or to obtain an approximate solution.
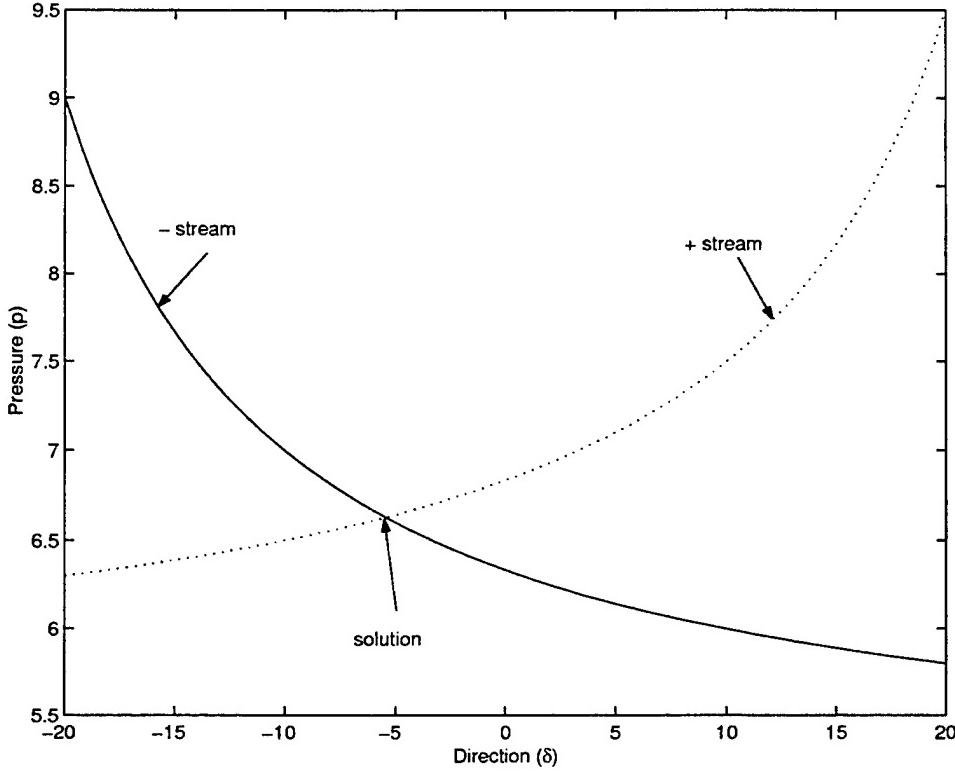
Figure 10. Pressure-Direction Curves Depicting the Riemann Problem

In this study, the complete solution was used. The setup of the approximate Riemann problem is thoroughly explained in [Ref. 22, 28].

The Riemann problem is solved by computing the final flow direction i.e. the direction that generates identical final pressures in both streams. The initial flow directions, δ_- and δ_+, and the initial pressures, p_- and p_+, are related to the final direction and pressure, δ_f and p_f, by the shock and expansion equations given below. The first equation, the shock equation, is valid for $p_f > p_\pm$, and the second equation, the expansion equation, holds for $p_\pm > p_f$.

$$\delta_f = \delta_\pm \pm \arctan\left(\frac{p_f - p_\pm}{(\gamma M_\pm + 1)p_\pm - p_f}\right)^2 \sqrt{\frac{p_\pm(2\gamma M_\pm^2 - \gamma + 1) - (\gamma + 1)p_f}{(\gamma + 1)p_f + (\gamma - 1)p_\pm}}$$

(II.55)

$$\delta_f = \delta_\pm \pm \nu(M_\pm) \mp \nu(M_f)$$

40

where

$$M_f = \frac{2}{\gamma - 1} \sqrt{\left(\frac{p_\pm}{p_f}\right)^{\frac{\gamma - 1}{\gamma}} \left(1 + \frac{\gamma - 1}{2} M_\pm^2\right) - 1}$$

(II.56)

$$\nu(M) = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \arctan \sqrt{\frac{M^2 - 1}{\left(\frac{\gamma + 1}{\gamma - 1}\right)}} - \arctan \sqrt{M^2 - 1}$$

and $\delta_\pm$ and $p_\pm$ represent one of the flow directions and pressures. To solve the shock and expansion equations for both flow directions, an iterative process is used. For the $n + 1$ iteration, two points are known on each of the curves shown in Figure 10. For the - stream curve, the points (δ_-^n, p_-^n) and $(\delta_-^{n-1}, p_-^{n-1})$ are known. A line is drawn through these points on each curve. The intersection of this line serves as the estimated pressure and is substituted into (II.55) as p_f. Similarly, p_-^n and δ_-^n are substituted for $p_\pm$ and $\delta_\pm$. Making these substitutions allows for the evaluation of δ_{f-}. Substituting the values from the + stream curve results in δ_{f+}. The iteration is terminated if $|\delta_{f+} - \delta_{f-}| < 10^{-3}$.

Before using the solution from the Riemann problem, the angular orientation of the cell edge must be computed. This direction is given by

$$\theta = \arctan -\frac{\mathbf{n}_{z_{n+\frac{1}{2},m}}}{\sqrt{\mathbf{n}_{x_{n+\frac{1}{2},m}}^2 + \mathbf{n}_{y_{n+\frac{1}{2},m}}^2}}$$

(II.57)

where the subscripts on $\mathbf{n}$ describe the direction of the normal vector from the cell face.

With the cell edge orientation known, the properties on the cell edge can be calculated. The pressure and density are found by using the Riemann solution in the θ direction. The magnitude of the velocity is computed using the results from the Riemann solution

$$M_\theta = \sqrt{1 + \frac{\gamma + 1}{\gamma - 1} \tan \frac{\pi}{2} + \nu(M_f) \pm \delta_f - \theta \sqrt{\frac{\gamma + 1}{\gamma - 1}}}$$

(II.58)

41

Converting this value to a dimensional value, q_θ, the velocity components can be determined

$$u_e = \frac{q_\theta \sin(\delta)\mathbf{n_x} - v_t\mathbf{n_y}}{|(\mathbf{n_x}, \mathbf{n_y})|}$$

$$v_e = \frac{q_\theta \sin(\delta)\mathbf{n_y} + v_t\mathbf{n_x}}{|(\mathbf{n_x}, \mathbf{n_y})|} \tag{II.59}$$

$$w_e = q_\theta \cos(\delta)$$

where v_t is the velocity component tangent to the cell edge described by

$$v_t = \frac{(-\mathbf{n_y}, \mathbf{n_x}) \cdot (u, v)}{|(\mathbf{n_x}, \mathbf{n_y})|} \tag{II.60}$$

The cell center from which v_t is determined is based on the relative sizes of θ and δ_f.

$$v_{t_{n+\frac{1}{2},m}} = \begin{cases} v_{t_{n+1,m}} & \text{if } \theta \geq \delta_f \\ v_{t_{n,m}} & \text{if } \theta < \delta_f \end{cases} \tag{II.61}$$

Having the properties at the cell edges, the corrector values are determined. The cell edge properties are substituted into the Euler equation in conservation form (II.36) and the fluxes are evaluated. The flux evaluations are decoded giving the final flow properties at the cell side at $z + \Delta z$.

$$u_{n,m} = \left(\frac{u_3}{u_1}\right)_{n,m}$$

$$v_{n,m} = \left(\frac{u_4}{u_2}\right)_{n,m}$$

$$w_{n,m} = \frac{\gamma}{\gamma + 1}\left(\frac{u_2}{u_1}\right)_{n,m} \sqrt{\left|1 - \frac{\gamma^2 - 1}{\gamma^2}\chi\right|} \tag{II.62}$$

$$p_{n,m} = \frac{(u_2 - wu_1)_{n,m}}{A_{n,m}}$$

$$\rho_{n,m} = \frac{u_{1_{n,m}}}{w_{n,m}A_{n,m}}$$

where $A_{n,m}$ is the cell face area defined in Section 3 and

$$\chi = \left[\frac{2H_0 u_1^2 - u_3^2 - u_4^2}{u_2^2} \right]_{n,m} \tag{II.63}$$

and u_i corresponds to the ith row of $U^k_{n,m}$ (see (II.37)).

7. Boundary Conditions

Discussion of boundary conditions for this scheme has thus far been conspicuously absent. The boundaries in ZEUS are identical to those in CM3DT, specifically, the body surface and the bow shock wave. However, since ZEUS uses a finite volume scheme, no grid points lie on the boundary; cell edges lie along the boundary. The properties along these edges are based on extrapolation from the center point. It should be noted that a modified version of Godunov's technique is applied to edges adjacent to the body surface. This scheme uses one sided differences to approximate derivatives normal to the surface. Additionally the flux passing through cells adjacent to the body is constrained by the tangent flow boundary condition. A full description of this modified version is in [Ref. 22].

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODELING THE PROJECTILES

A. BACKGROUND

As with any simulation, the accuracy of the results is based on the accuracy of the input. With this in mind, it was imperative to accurately portray the shapes of the four projectiles. Both CM3DT and ZEUS accept user generated geometry files in Plot3D (P3D) format. In this format, grid points aligned on axial (cross-sectional) and circumferential planes describe the surface and each point is stored as an (x, y, z) coordinate. MATLAB computer codes, *h4generator.m, h4agenerator.m, h5generator.m, h5agenerator.m*, were designed to construct each projectile in P3D format. In each code, the points were output to a ASCII file in Plot3D readable form. The general format for loading points into this form is

WRITE((X(i,j),i=1..imax),j=1..jmax)
WRITE((Y(i,j),i=1..imax),j=1..jmax)
WRITE((Z(i,j),i=1..imax),j=1..jmax)

where the i represents the axial direction and j represents the circumferential direction as shown in Figure 11. These generation codes are located in Appendix A.

The H4 was easily defined mathematically and so constructing an accurate model was simple. Similarly, the H5, being a smooth surface, was also relatively easy to construct. The projectiles with strakes offered the greatest challenge. First, modifications to the shape had to be made to make it usable in ZEUS. Then, a mathematical method for describing these shapes had to be developed. This chapter will discuss the construction of the geometry files defining the four projectiles. It should be noted that because of pitch plane symmetry associated with pitch only flight, the models used in these tests consisted of only half of the projectile (positive x).
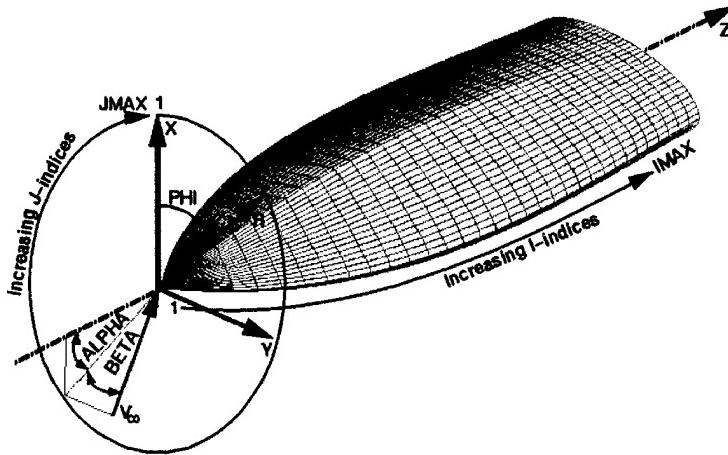
Figure 11. Notation of Indices for Creation of Plot3D Files

B.   H4

The H4 is easily defined in ZEUS Cartesian coordinates (see Figure 11) as

$$25x^2 + 9y^2 = 15z \qquad 0 \le z \le 540 \qquad \text{(III.1)}$$

where $z = 0$ is at the nose and $z = 540$ is the value at the tail. By applying the standard definition of an ellipse in parametric form

$$\begin{aligned} y &= a \cos \phi \\ x &= b \sin \phi \qquad 0 \le \phi \le 2\pi \end{aligned} \qquad \text{(III.2)}$$

then (III.1) becomes

$$25b^2 \sin \phi^2 + 9a^2 \cos \phi^2 = 15z \qquad \text{(III.3)}$$

To determine the magnitudes of the semi-major and semi-minor axes, b and a, respectively, the conditions $\phi = 0$ and $\phi = \dfrac{\pi}{2}$ are applied. At $\phi = 0$

$$\begin{aligned} 9a^2 &= 15z \\ a &= \frac{\sqrt{15z}}{3} \end{aligned} \qquad \text{(III.4)}$$

At $\phi = \dfrac{\pi}{2}$

$$\begin{aligned} 25b^2 &= 15z \\ b &= \frac{\sqrt{15z}}{5} \end{aligned} \qquad \text{(III.5)}$$

46

Using a direct mapping of z to ζ, the H4 is parametrically defined as

$$y = a \cos \phi$$
$$x = b \sin \phi \qquad \text{(III.6)}$$
$$z = \zeta$$

where a and b are defined above and

$$0 \leq \zeta \leq 540$$
$$0 \leq \phi \leq 2\pi \qquad \text{(III.7)}$$

From these equations, the model was generated using a MATLAB code. This algorithm modeled the H4 (half shell) with 60 circumferential points (x and y) coordinates and 109 axial (z) coordinates for a total of 6,540 points. Since the H4 serves as the basic shape for each of the other projectiles, the derivations noted above will prove crucial in modeling the remaining objects.

C.   H5

Like the H4, the H5 was smooth and relatively easy to define mathematically. For $0 \leq \zeta \leq 430$, the H4 and H5 are identical. Therefore, this section will discuss the modeling of the flared tail that distinguishes the H5.

The flare of the H5 was designed as an elliptical conic section. The ellipse that describes the tail of the flare has a different eccentricity than that of the H4. Therefore, the flare was not easily defined mathematically. But because the base of the flare was described by a known equation, then for every ϕ, the (x, y) coordinates could be determined for both the $z = 430$ and $z = 540$ cross flow planes. At $z = 430$, the beginning of the flare, these points were computed using (III.6). The tail of the flare was an ellipse defined by

$$22.85^2 y^2 + 35^2 x^2 = 799.75^2 \qquad \text{(III.8)}$$

47

Using the same method used in the previous section, the semi-minor and semi-major axes were found

$$a = \frac{799.75}{22.85} = 35$$

$$\text{(III.9)}$$

$$b = \frac{799.75}{35} = 22.85$$

Having determined these constants, the parametric equation for the tail plane became

$$y = 35\cos\phi$$
$$x = 18\cos\phi \qquad 0 \le \phi \le 2\pi$$

$$\text{(III.10)}$$

For every specified ϕ, points $(x, y, z)_{z=430}$ and $(x, y, z)_{z=540}$ were determined using (III.6) for $z = 430$mm and (III.10) for $z = 540$mm. From the pair of coordinates $(x, y, z)_{z=430}$ and $(x, y, z)_{z=540}$ in a given ϕ plane, the equation of a line was derived and equally spaced grid points along this line were computed. This process was performed for each circumferential plane. Like the H4, the modeled H5 had 109 axial grid points (of which 21 defined the flare) and 60 circumferential points.

D. H4A

The projectiles with strakes offered two problems not found in the projectiles without strakes. First, the bodies were no longer smooth and could not be simply defined by mathematical equations. Even using a grid generator, the design of the H4A and H5A was cumbersome. Second, the angles between strake sides and the body were potential problems for the Euler solver, particularly on the shoulder side of each strake. ZEUS, like other CFD codes, relies on the orthogonality of the grid to obtain an accurate solution. In fact, ZEUS was unable to properly "march" axially; negative pressure values occurred in these areas. To overcome this problem, the shape of the strakes had to be modified while maintaining an accurate depiction of the true shape of the projectiles. A "smoothing" of each strake's shoulder side was performed. This modification was minor and resulted in only a 2% increase in the cross-sectional

48

area at the tail. Figure 12 shows the difference between the actual and modeled strake in a cross flow plane.
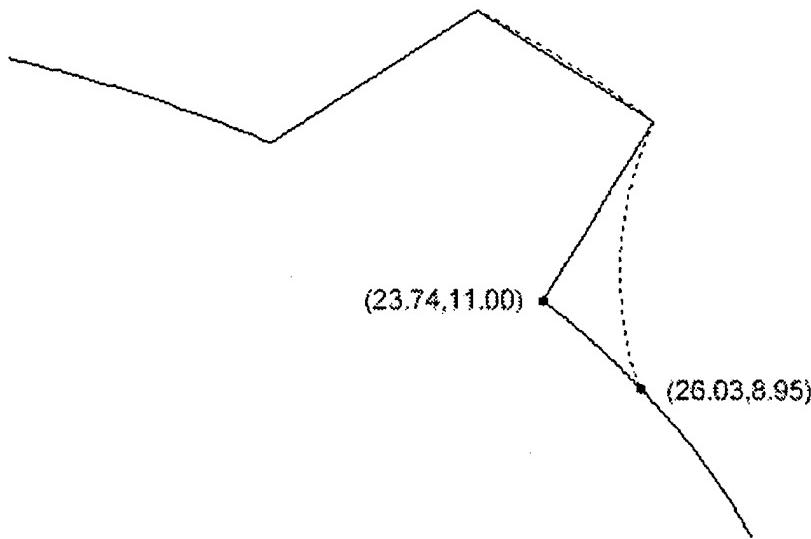
Figure 12. Actual and Modeled Strake

In keeping with the methods used to model the H4 and H5, the H4A was also modeled circumferentially. Specifically, the projectile was modeled in five pieces: the body from $\phi = 0^\circ$ to the upper strake ($\phi = 0^\circ$ in ZEUS coordinates is the x-axis); the strake; the body between the upper and lower strakes (the shoulder of the projectile); the lower strake; and the body from the lower strake to $\phi = -180^\circ$. Because the strakes followed the ϕ planes, this was the easiest method of modeling the projectile. The body points were still easily defined by (III.6). The strakes were constructed by defining each side: the leeward or windward side; the top; and the shoulder side.

The leeward and windward sides were modeled exactly, that is, no smoothing was used. The angle between these strake sides and the body allowed a grid to be generated and ZEUS was able to march across these areas.

The top of the strakes were slightly curved instead of being squared off as designed. The equation for the flare tail, (III.10), was used. The reason for this will be discussed in the next section.

Figure 12 shows the sharp edges caused by the strakes. The strake sides closest to the shoulder of the projectile have significantly steeper slopes than the leeward and windward sides of the strakes. These steep slopes result in smaller angles between the strake and the body. In an attempt to alleviate this problem, the strake sides were smoothed. A comparison between the actual and modeled strake side is depicted in Figure 12. The coordinates show the difference between the actual and modeled strake-body intersection points.

In two dimensions (the cross flow plane), the strake sides are straight lines as shown in Figure 12. Because the strakes taper, the length and slope of these lines change in each cross flow plane. That is, moving toward the nose, these lines become shorter and have decreasing (in absolute magnitude) slopes. To smooth the entire side, the lines in each cross flow plane were represented as circular arcs. The arc sections in each cross flow plane were defined by a circle whose center and radius were dependent upon the axial station. To implement this method, three points defining each circle were required. The endpoints of each arc, A and B in Figure 13, were used. It is important to note that B, the point where the strake and body intersect, is not the exact location of intersection; point C describes where the actual strake intersects the body. Point B was chosen by selecting an arbitrary distance from point C, approximately 4mm. To compute a third point, thereby uniquely defining the parent circle of the arc, an approximation based on midpoints was used.

The midpoints of segments $\overline{AB}$ and $\overline{AC}$, denoted E and D respectively, were found. The midpoint of the segment connecting these points, F, was the third point. Now an equation for the circle defined by the points, A,B, and F, was found by

solving

$$r = \sqrt{(x_A - h)^2 + (y_A - k)^2}$$
$$r = \sqrt{(x_B - h)^2 + (y_B - k)^2} \qquad \text{(III.11)}$$
$$r = \sqrt{(x_F - h)^2 + (y_F - k)^2}$$

where r is the circle radius and (h, k) is the center. Once the equation of the circle was derived, an arbitrary number of grid points were computed to represent the arc. This process was repeated for each cross flow plane.
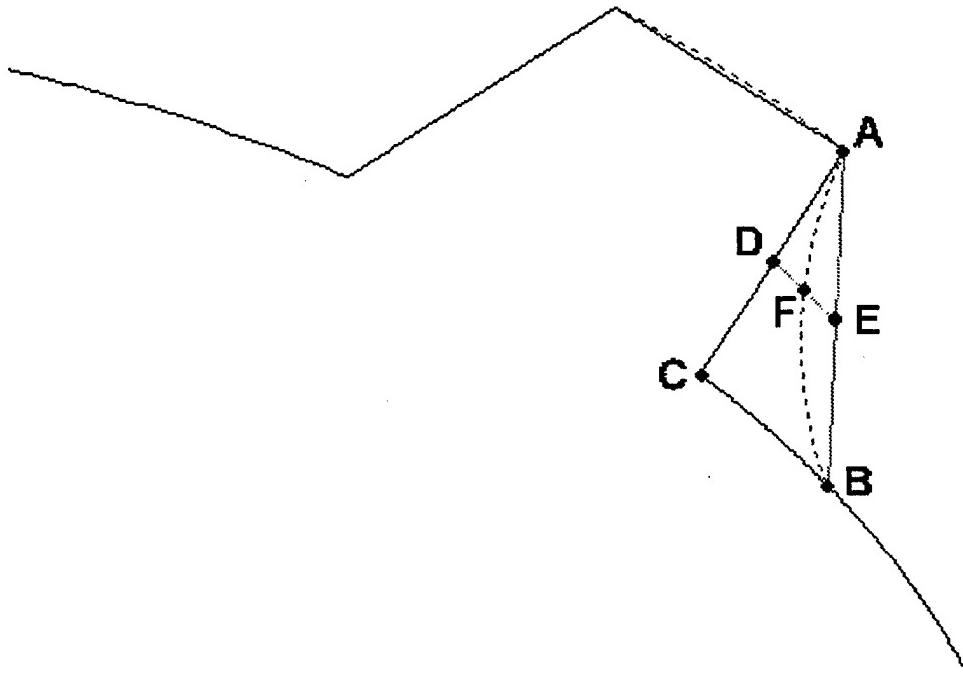
Figure 13. Construction of the Modified Strake Side

As with the previously modeled projectiles, 109 axial coordinates were used. However, circumferentially, 61 coordinates were used. Additionally, the circumferential planes were not uniformly distributed. To capture the smoothing on the shoulder sides and tops of the strakes, points were clustered in these areas.

E.   H5A

The construction of the H5A model was a combination of the processes of modeling the H5 and the H4A. However, when the flare begins at $z = 430$, the

strakes begin to blend into the body. But as previously mentioned, the strakes follow a constant ϕ. Therefore, the methods used to compute the sides of the strakes were identical to those used in the development of the H4A with different strake-body intersection points. These points were determined using the flare building algorithm of the H5.

It was mentioned previously that the strake top was slightly curved (virtually impossible to detect visually) instead of the squared top in the design. Because the strake is supposed to blend completely into the flare at $z = 540$, the curved top made this transition easy to design with negligible aerodynamic effects.

Figure 14 depicts the modeled projectiles.
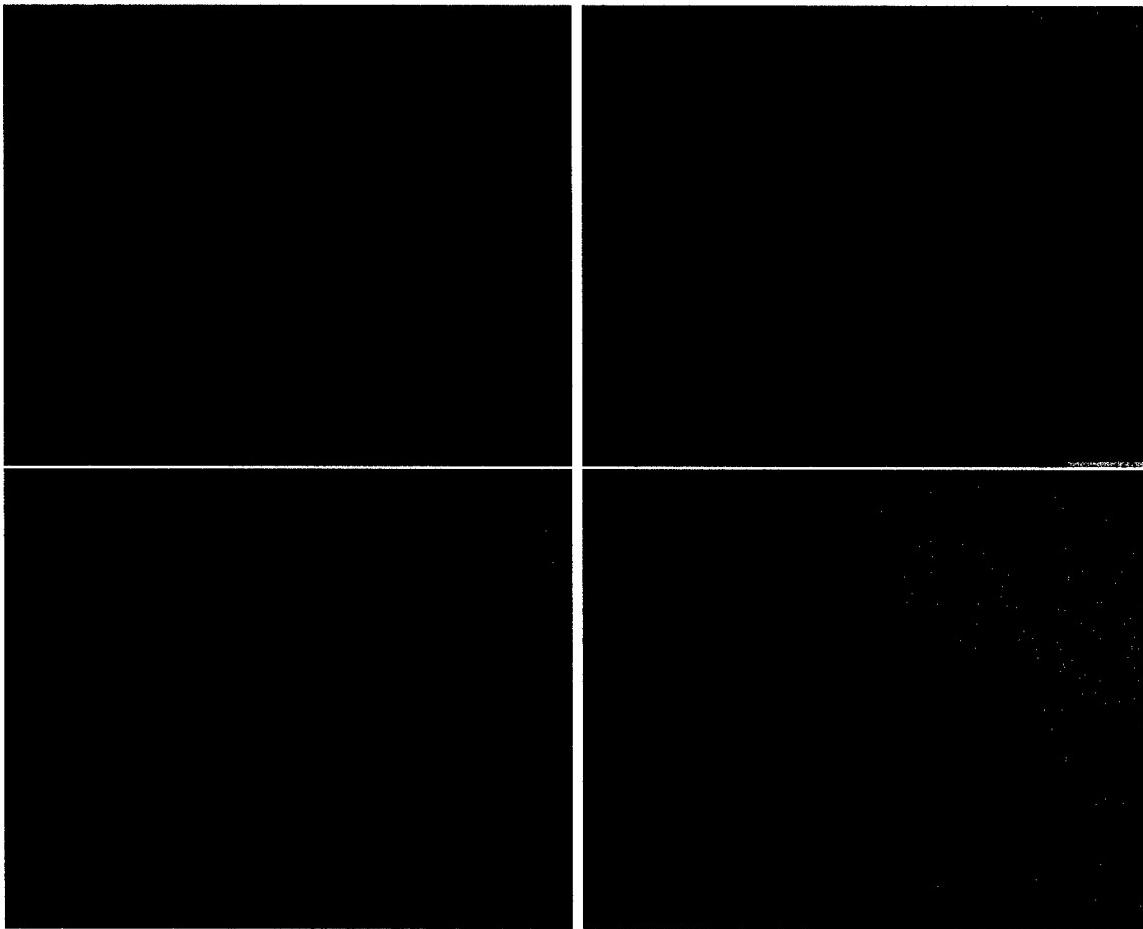
Figure 14. Models of the H4 (top left), H5 (top right), H4A (bottom left), H5A (bottom right)

F. PROJECTILE NOMENCLATURE

Having defined and named the projectiles, it should be noted that previous works have used other nomenclature to define these projectiles. For clarity, in this work, projectiles with strakes are indicated by the "A" identifier and application of the flare is denoted by the numerical identifier. Table I shows the nomenclature used in previous works.

Description	Present Nomenclature	Prior Nomenclature
Elliptic Projectile	H4	H3
Projectile with strakes only	H4A	H4A
Projectile with flare only	H5	not named
Projectile with strakes and flare	H5A	H5

Table I. Present and Prior Projectile Nomenclatures

THIS PAGE INTENTIONALLY LEFT BLANK

IV.   APPLYING THE CM3DT CODE

A.   BACKGROUND

Since the H5/H5A differ from the H4/H4A only in the tail flare, nose solutions were required for only two geometries: the projectile with strakes and the projectile without strakes. That is, the nose solutions for the H4 and H4A were used for the H5 and H5A respectively. Since the method of obtaining the nose solutions was nearly identical for both geometries, the discussion applies to both nose configurations except where noted. To avoid confusion, the ZEUS coordinate system will be used when describing objects in this chapter.

The application of the CM3DT code is actually a three step process involving manipulation of the model geometry files, the use of a geometry pre-processor, and then the running of the solver.

In this chapter, the methodology, user inputs, and code output pertaining to the nose tip solution will be discussed. As noted previously, only information pertinent to this specific problem will be addressed. Additional information on the application of the CM3DT code to other problems can be found in the CM3DT user guides [Ref. 25, 26].

B.   PREPARATION OF THE GEOMETRY INPUT FILES

To prepare the geometry file to be input into CM3DT code, four operations had to be performed. First, since the CM3DT is concerned only with nose solutions, the nose of each projectile had to be parsed from the afterbody. Unlike many conventional (cone-cylinder) projectile shapes, the noses of the projectiles in this study were not clearly defined. In this research, the length of the nose was 5mm. To ensure all required geometric data was obtained, the first 10mm of the projectiles was used. The size was based on the desired location (output) of the solution. This will be discussed in further detail later in this chapter.

Next, the number of points defining the nose was rescaled. CM3DT limits the number of point to 162. Specifically, only nine circumferential and 18 cross-sectional planes can be used. The models described in the previous chapter inadequately define the nose cross-sectionally (only three planes instead of the 18 in the first 10mm) and over-define in the circumferential direction (over 50 planes). Additional problems existed.

The version of CM3DT used in this study was modified specifically for use with ZEUS. However, one major idiosyncrasy remains; the ZEUS and CM3DT coordinate systems are different. The orientation of the projectile had to be modified to align with the CM3DT coordinate system. Both computer programs read standard P3D geometry files that provide grid points in Cartesian coordinates, however the orientation of the axes is different. The ZEUS coordinate system defines x as up, y to the right, and z running axially aft. Conversely, CM3DT defines y as up, z to the left, and x running axially aft. Figure 15 compares the two different coordinate systems. Since the majority of the calculations on these projectiles occurred in ZEUS, they were modeled in ZEUS coordinates as discussed in Chapter III.

Finally, CM3DT failed (specifically the pre-processor) when the nose tip was the origin. The origin caused a singularity and the nose tip had to be modified. The singularity occurs when transforming from Cartesian coordinates to the cylindrical-like system described in Chapter II; this transformation requires the evaluation of $\frac{1}{R}$. A simplistic modification was made i.e. eliminate the origin. The nose tip of each projectile was "shaved" resulting in the nose tip being described by an extremely small plane instead of a point, the origin. In ZEUS coordinates, the nose tip is a plane at $z = 10^{-10}$ instead of the origin. The method for overcoming this problem was provided by Edge [Ref. 30].

Although any of these problems left unmodified would cause catastrophic results, the required changes were easily implemented. The *h4generator.m* and *h4agenerator.m* codes were modified to create two nose generating codes *h4nosegenerator.m*
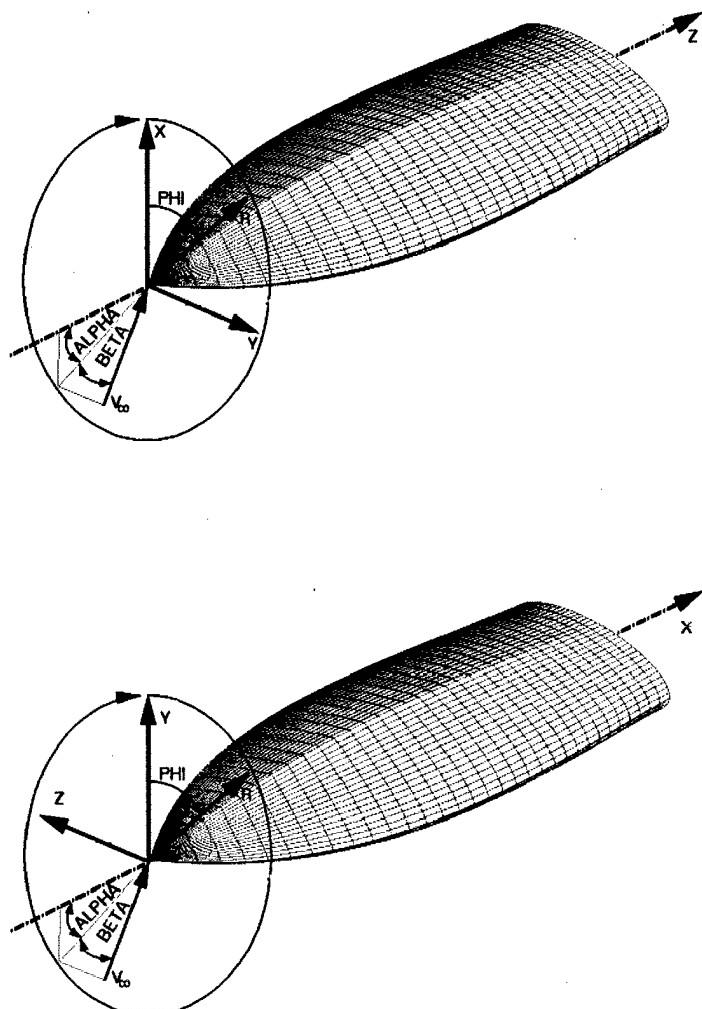
Figure 15. ZEUS (a) and CM3DT (b) Coordinate Systems

and *h4anosegenerator.m*. The modifications were simple and accomplished four goals: to parse the nose from the projectile; to redefine the number of planes (grid points) describing each nose; to map the objects into the CM3DT coordinate system; and to remove the singularity caused by the origin.

To parse the nose and redefine the grid points describing the nose, each z vector was changed from a vector of length 109 going from 0 to 540mm to a vector of length 18 going from 10^{-10} to 10mm. Similarly, the number of circumferential planes was reduced to nine. To transform each point into the CM3DT coordinate system, the output of the generators was modified. In their original (ZEUS) form, the code outputs grid points as shown in Chapter III. To transform into CM3DT coordinates, the data was written to the geometry file in this format

$$\text{WRITE}((\text{Y}(i,j),i=1..imax),j=1..jmax)$$
$$\text{WRITE}((-\text{Z}(i,j),i=1..imax),j=1..jmax)$$
$$\text{WRITE}((\text{X}(i,j),i=1..imax),j=1..jmax)$$

where the indices are shown in Figure 11.

Special treatment was used on the nose with strakes. Because of the limited number of circumferential planes (or points in each cross-sectional plane), the nose with strakes could not be accurately modeled; at least eleven points are needed. Using the nine points available, an approximation was constructed. In this model, the strakes are pointed instead of squared (see Figure 16).

C.   GEOMETRY PRE-PROCESSOR

Once the modifications to the geometry were completed, a pre-processing code, *preproc.x* was run. This code, part of the CM3DT software package, simply converts the P3D file to a readable form for CM3DT. The Cartesian coordinates are converted to the cylindrical-like system (x, y, ϕ) described in Chapter II. When this code was run, the inputs in Table II were used in the required input file *preproc.inp*. It should be noted that the "1" usually found on the first line of a "multi-grid" formatted P3D
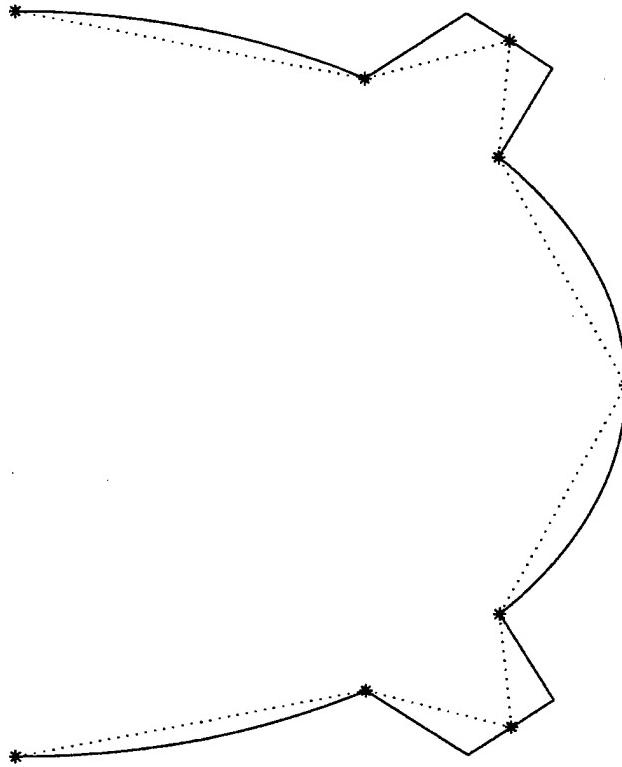
Figure 16. Actual (Solid) and Approximated (Dotted) Straked Nose Cross-Section

file (the format accepted by ZEUS) must be removed prior to execution of the code. CM3DT uses the "single-grid" P3D format which omits this line. In the codes shown in Appendix A, the "1" is written to the output files. Visualization programs require this number in P3D files.

The output from the preprocessor is *cm3dt.geo*, a file that is read during the execution of the next step of the process.

Variable	Description	Value
isa	Pitch plane symmetry (1=yes)	1
lmax	Number of circumferential planes	9
fname	Name of the file to be processed	*h4nose.grd*

Table II. Geometry Pre-Processor Input Variables

D. THE FLOW FIELD SOLVER

1. Required Inputs

The solver, *Cm3dt2.x*, requires two input files: *cm3dt.geo* (pre-processor output) and *CM3DT.INP*. The *CM3DT.INP* file consists of the flow conditions and other information described in Chapter II. The variables and values used in this research are shown in Table III. The variables below the solid line are variables unique to the ZEUS-coupled version of this solver. An example of *CM3DT.INP* and *preproc.inp* are in Appendix B. At this point, the choice of some of these values deserves explanation.

Variable	Description	Value
AMINF	Mach number	5-8.2
alpha	Angle of attack	0-15
beta	Angle of sideslip	0-15
ILAM	Use of λ-differencing scheme (1=yes)	1
KMAX	Number of time steps	1000
NMAX	Number of grid points between surface and shock	11
MMAX	Number of grid points along each circumferential plane	18
LMAX	Number of circumferential planes	9
KOUT	Number of time steps between complete output	1000
R_N	Effective nose radius	.5
XBO	Location of nose tip in z direction (ZEUS coordinates)	0
XBB	Points in z direction (ZEUS coordinates) describing downstream boundary	9*8
NA	Number of cells between edges 1 and 3 in ZEUS	18
MA	Number of cells between edges 2 and 4 in ZEUS	24
ZSTART	Axial location where starting data is required	5
Z0	Location in ZEUS coordinates of X_0 in CM3DT coordinates	0
X0	Location in CM3DT coordinates of Z_0 in ZEUS coordinates	0

Table III. CM3DT Input Variables

The number of time steps, **KMAX**, was chosen based on a recommendation from Edge [Ref. 29]. This value had to be sufficiently large so that the convergence criteria described in Chapter II could be met.

The density of the solution grid, described by **NMAX**, **MMAX**, and **LMAX**, was maximized to obtain the highest resolution possible.

The effective nose radius, R_N, is used to estimate the shock stand-off distance and the location of the hinge points. For blunt objects, the bow shock is separated from the object. This shock stand-off distance is based on the bluntness of the object. The default value of this variable was used. The value selected represents a reasonable stand-off distance and allows the hinge points to accurately simulate the contour of the body surface. The selection of the value of R_N is arbitrary, because of the iterative nature of this solver. In theory, convergence will occur for any value of R_N. However, a reasonable estimate speeds the solution process.

XBB describes the downstream boundary i.e. the last axial plane for which CM3DT will provide a solution. Two requirements are imposed on this value. It must be less than the length of the nose and it must be larger than the location of the desired starting data. Because the solution is initially aligned with the body and therefore not a $z =$constant plane, the solution at this boundary cannot be used for a ZEUS initial flow condition. A built-in interpolation algorithm is used to obtain a solution field for a $z =$constant plane. A tenable value of **XBB** is determined by the shape of the projectile and the desired location for the ZEUS starting data. It is also important to note that CM3DT is accurate for nose solutions and not the afterbody; its utility in this study is solely to provide initial data for the afterbody solver. Therefore, the value of **XBB** was selected to accurately represent the shape of the nose yet allow ZEUS to provide the flow field solution for the majority of the projectile. Additionally, the value of **XBB** had to be sufficiently large to allow interpolation of flow conditions on the $z =$**ZSTART** plane.

Of the inputs required for the ZEUS coupled version of CM3DT, **NA, MA,** and **ZSTART** are of interest. **NA** and **MA** describe the density of the output grid. Because of a conversion process used in ZEUS that will be discussed in the following chapter, any reasonable choice for these values can be used. As mentioned above, the value of **ZSTART** is chosen to obtain a sufficient representation of the projectile's nose while allowing ZEUS to solve for the afterbody flow field. Since these projectiles

do not have a clearly delineated nose, the choice of this value was arbitrary. A decision was made to keep the length of the nose below 1% of the projectile length. Since the output at this value is only serving as initial solution for the marching solver, the location of the solution was approximated at one percent of the projectile length.

2. Output

The execution of *Cm3dt2.x* results in a text output file, (*CM3DT.OUT*), a ZEUS-ready input file (*ZEUS.L03*), and a series of graphical output files (*CM3DT.L***). Of the three, *CM3DT.OUT* and the graphics files are of interest; *ZEUS.L03*, although critically important in the running of ZEUS, was in an unreadable format to the user. The graphic output served as a common sense check of the solution. Using FAST and a script written by Edge [Ref. 5], the solution was graphically depicted. As an example, Figure 17 depicts the solution (pressure contours) for the H4 at 0° angle of attack at Mach 5. From this graphic output, an inappropriate grid, poorly shaped shock,
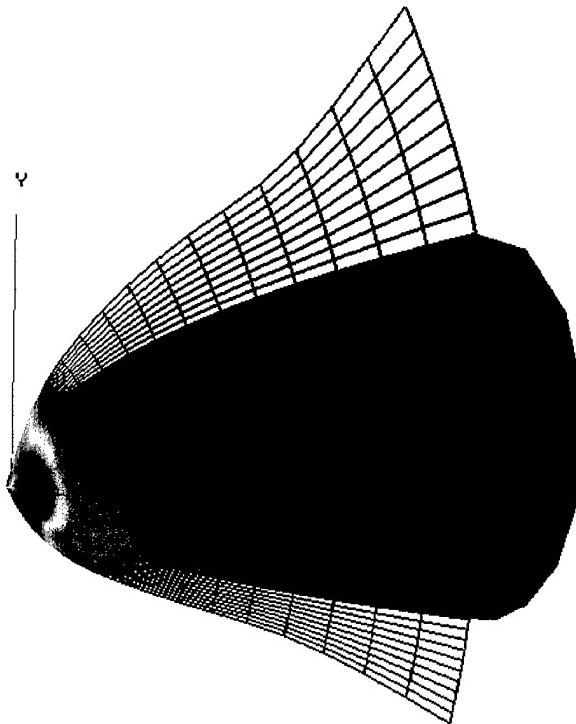
Figure 17. Example of CM3DT Graphic Solution Output

or other anomalies such as unexpected pressure changes were easily identifiable. The text output file provided many more details and, in essence, walked the user through the process. This file was particularly useful for troubleshooting. For example, an inconsistency identified in the graphic output could be isolated and examined in the test output.

Finally, *ZEUS.L03* was the critical link between CM3DT and ZEUS. This file contained the values of p, ρ, u, v, and w at each grid point as well as bow shock speeds and locations in the $z = $ **ZSTART** plane. This information is packaged in a format that can be read by ZEUS.

THIS PAGE INTENTIONALLY LEFT BLANK

V. APPLYING THE ZEUS CODE

A. BACKGROUND

Having obtained an initial solution plane, the afterbody was analyzed using ZEUS. Application of the ZEUS code was greatly facilitated by the use of the ZEUS Graphical User Interface (GUI) created by Edge and Clarke [Ref. 5]. The ZEUS GUI offered three advantages over normal command line application:

- easy input file manipulation
- an ability to observe the computational grid prior to attempting a solution run
- direct links to FAST and pre-written scripts enabling the user to visualize the computational grid and the solution

This section will discuss the application of the ZEUS GUI in obtaining solutions for the projectile afterbodies. The methodology, inputs, and output will be presented. Except where noted, the discussion in this chapter applies to all cases.

B. INPUT FILES

Wardlaw and Davis [Ref. 27] placed all input variables into one file. To facilitate the development of the GUI scripts, Edge and Clarke [Ref. 5] split the ZEUS input files into six categories:

- initial conditions
- integration control
- zone definitions
- boundary conditions
- geometry definitions
- output control

The specific inputs from each of these categories will be discussed in this section. An example of these input files can be found in Appendix B.

65

1. Initial Conditions

The initial condition category describes the freestream conditions used in this study. As previously mentioned, ideal gas conditions were applied to this case. The inputs used are shown in Table IV.

Variable	Description	Value
ALPHA	Angle of attack	0-15
BETA	Angle of yaw	0
XMINF	Mach number	6-8.2
PINF	Pressure	1
DINF	Density	.00001

Table IV. Initial Condition Input Variables

2. Integration Control

The integration controls are used in this code to specify the numerical constraints that the code uses. Included in this category are variables describing the axial station at which ZEUS ends it calculation in addition to the CFL safety factor that determines the marching step size based on a percentage of CFL value. For runs where no modifications to the grid or zoning schemes were made, the length of each projectile was used for **ZETAEND**. The step size safety factor value used was the default value (90%); the use of this value was based on previous work performed on similar objects [Ref. 4, 15]. Decreasing the value of **FCFL** significantly increased the number of steps required to achieve a solution. The maximum number of steps, **KEND**, serves only as a safety in this study. The number of steps required to achieve a solution varied based on the complexity of the projectile shape, the angle of attack, and the velocity. The H4 at $\alpha = 0^o$ and Mach 5 required less than 900 steps. However the H4A at $\alpha = 15^o$ and Mach 8.2 required over 10,000 steps. In most cases, the number of steps required to obtain a flow field solution exceeded maximum value allowed in the ZEUS GUI. Continuation of the process beyond the maximum number

of steps was performed by restarting ZEUS using the last known solution plane as an initial boundary condition, a process identical to that used with the nose solution.

In Chapter II, application of the limiter, K, in (II.46), was described. The input variable for this limiter, **XKI**, is used. Large values of this variable typically produce sharp shocks, but often lead to pressure oscillations. Conversely, small values result in less than second order accuracy and induce excessive shock smearing. Wardlaw and Priolo [Ref. 27] noted that the default value of 1, produced accurate results and that non-default values should be used to "revive" a faulty computation. Finally, one input not listed in the Table V, **ZSTART**, was required by ZEUS; this input, like many others, was read by ZEUS from the solution file from CM3DT, *ZEUS.L03*. The value of this variable was specified in *CM3DT.INP*. The values used are shown in Table V.

Variable	Description	Value
ZETAEND	z value for calculation termination	540
KEND	Maximum steps	3000
FCFL	Step size safety factor	.9
XKI	Limiting constant	1
DFAC	Turn angle multiplication factor	1

Table V. Integration Control Input Variables

3. Zone Definition Input Variables

The variables in this input file are one of two files used by ZEUS to construct the computational grid. The values chosen for this input file were based on work by Priolo and Wardlaw [Ref. 15]. They successfully applied ZEUS to elliptic shaped projectiles (2:1, 2.5:1, 3:1 ellipticity ratios) using a single-zoned 72x72 mesh and a singled-zoned 36x36 mesh. Use of a denser mesh was chosen to provide greater resolution particularly on the projectiles with strakes. Because the strakes were modeled as part of the body, only a single zone was used. Use of multiple zones in ZEUS are typically applied when fins are added to the body. Although use of this feature was

67

originally considered for this problem, the modeling of the projectiles as a body-alone object precluded the requirement for multiple zones. A potential use for additional zones is discussed in the grid sensitivity analysis in Chapter VI. The values and descriptions of the variables in this file are shown in Table VI. A graphic description of

Variable	Description	Value
IZN	Number of zones	1
NA	Number of cells in ξ direction	72
MA	Number of cells in η direction	72
MAZ(IZN)	Number of cells in η direction for each zone	72
INPANG	Zone spacing (equidistant=1)	1

Table VI. Zone Definition Input Variables

these variables is shown in Figure 18.
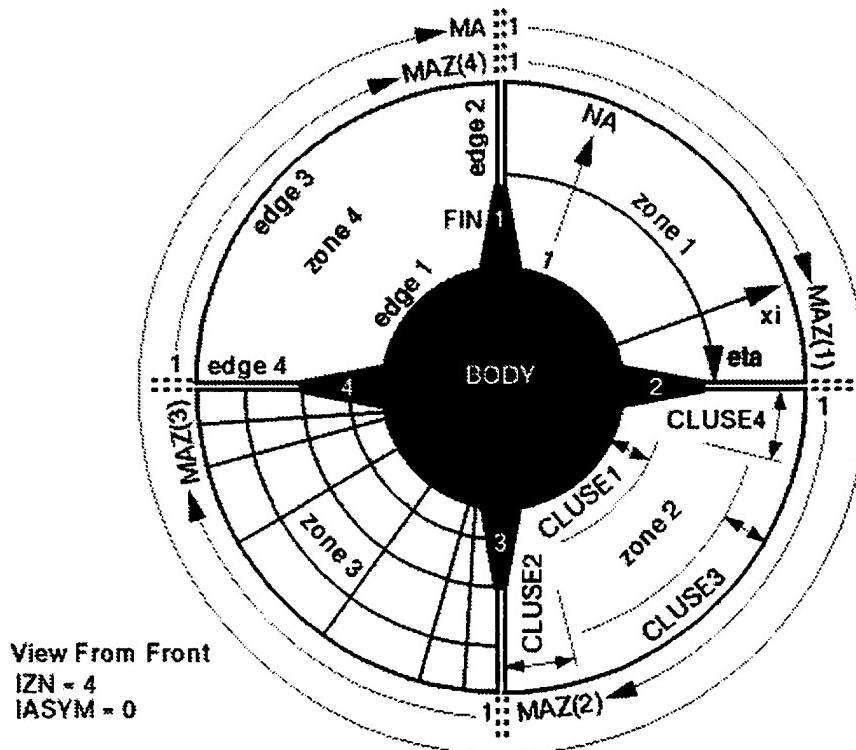
Figure 18. Zone Definition Variables

68

4. Boundary Conditions Input Variables

Based on the limitations of ZEUS, default boundary conditions are encoded in ZEUS expediting the input setup. Of interest are two input variables, **IASYM**, denoting pitch plane symmetry, and **ISHOCK**, which defines whether the outer boundary is to be shock fitted. Because the projectiles are symmetrical about the pitch plane, the pitch plane symmetry boundary condition was used. Based on slender body theory, it was assumed that all significant flow information was located between the body surface and the bow shock wave. Initial data required to perform the shock fitting was obtained from the nose tip solution in *ZEUS.L03*. The information in this file overrides an user input values from the GUI. The variables are listed in Table VII.

Variable	Description	Value
IASYM	Pitch plan Symmetry (yes=1)	1
ISHOCK	Edge 3 to be fitted by calculation of shock (yes=1)	1

Table VII. Boundary Condition Input Variables

5. Geometry Definition Input Variables

These variables provided the remaining information required to construct the computational grid. In addition, information regarding the geometry of the body was input through these variables. Of note in the values listed in Table VIII are those regarding grid clustering. The grid was clustered about the shoulder of the projectile. This was accomplished by doubling the spacing at the outer zone edges (2 and 4). By doing this, the grid at the shoulder was twice as dense of that at the lee and windside meridians. Clustering in the η direction was used to most accurately predict phenomena that would possibly occur at this location. Priolo and Wardlaw [Ref. 15] showed that ZEUS was not able to accurately compute the pressures at the shoulder even with a clustered grid. Fortunately, they showed that these inaccuracies had little effect on the overall solution.

69

Variable	Description	Value
ICORD	Type of coordinate system (cylindrical=1)	1
IMESHF	Mesh clustering in ξ direction (yes=1)	0
IMESHG	Mesh clustering in η direction (yes=1)	1
CLUSE4	Grid spacing at edge 4 boundary	2
CLUSE2	Grid spacing at edge 2 boundary	2

Table VIII. Geometry Definition Input Variables

6. Output Control Variables

The majority of the variables in this category were used to format the output and are described in [Ref. 5]. However, the reference area variable, **AREF**, was defined in this file. This area was used in the computation of the force and moment coefficients. The reference area used in this research was 21,600mm^2, the cross-sectional area of the tail of the H4. This is the convention used by Edwards and Roper [Ref. 10].

C. PREPARING THE CM3DT OUTPUT FOR USE IN ZEUS

Having created the required input files, the output from CM3DT had to be readied for use with ZEUS. *ZEUS.L03*, the solution output file from CM3DT, provided ZEUS with the flow properties at $z = 5$mm. However, this solution plane consisted of 18x24 grid points as defined in *CM3DT.INP* as **NAxMA**. The values at these points were used to interpolate values for the grid defined by *zondim.inp* and *geo.inp* using *convert.x*, part of the ZEUS package and integrated into the GUI. This code took the file *RESTART* (*ZEUS.L03* was copied to *RESTART*) and created a new initial plane aligned with the mesh characteristics (number of zones and cells, clustering ,etc.) defined by the user and wrote the output to the file *START*. Details on the interpolation algorithm used in *convert.x* is available in [Ref. 27]. A simple linear interpolation of the previous solution to a new grid is accomplished by re-defining the

70

number of grid points (**NA** and **MA**) and establishing new functions describing the clustering. The *START* file is the default file that ZEUS uses when the "Restart" option is selected in *intcntrl.inp*.

D.   EXAMINING THE COMPUTATIONAL GRID

The ZEUS GUI offers an extremely powerful tool called Pseudo-ZEUS. This program produced the solution grid within minutes (on a Silicon Graphics Octane). By producing the solution grid prior to computing a full (and potentially erroneous solution), the user was able to make time saving modifications. Since most full solutions required between 15 to 25 minutes, this program saved significant computer time by allowing errors to be corrected prior to the initiation of the full computational process.

Most problems were fixed prior to computation. However, one problem with the grid of the models with strakes was not corrected. On one side of each strake, orthogonality of the grid is poor. Figure 19 shows the skewness of the grid about this strake side. Additionally, only two cells define this side of the strake. Better orthogonality was achieved by dividing the body into multiple zones. This was performed for one case and the results are discussed in the grid sensitivity analysis in Chapter VI. An attempt to add more cells resulted in a worse grid structure and an improperly defined shock.

E.   ZEUS OUTPUT

Like CM3DT, ZEUS provided output in visual and numerical form. Typically, the visual solution was used for immediate feedback. FAST scripts in the ZEUS GUI allowed the user to automatically link into FAST. As an example, Figure 20 shows the pressure contours on the H5 at $\alpha = 15°$ at Mach 6.

The full text output in *echoexecut.txt* provided the velocity profile and the surface pressures. It also showed the ZEUS calculated force and moment coefficients

71
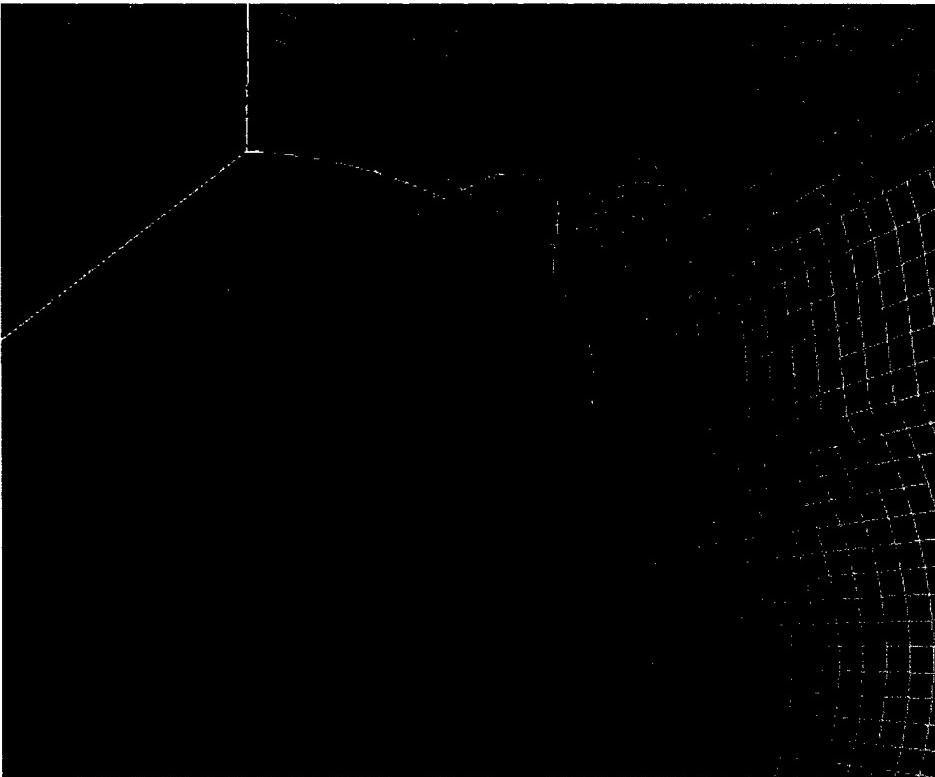
Figure 19. Grid Skewness About the Strake Side

at each axial station, the last being the total force or moment coefficient. Additionally, the center of pressure was provided. The output in *echoexecut.txt* was the source for the analysis in Chapter VI.

F.   A SPECIAL CASE: H4 AND H5 AT $\alpha = 15^O$, MACH 7 AND 8.2

In each of these four cases, a problem arose in prediction of the shock location and therefore the generation of the compuational grid. When the process mentioned in the previous sections was applied in these cases, the shock (and grid boundary) was severely deformed with an apparent oscillation about the shoulder of the projectiles. This problem is shown in Figure 21.

Investigating the problem revealed that the grid was highly clustered on the windward side. Although this grid density caused no problems at lower Mach num-
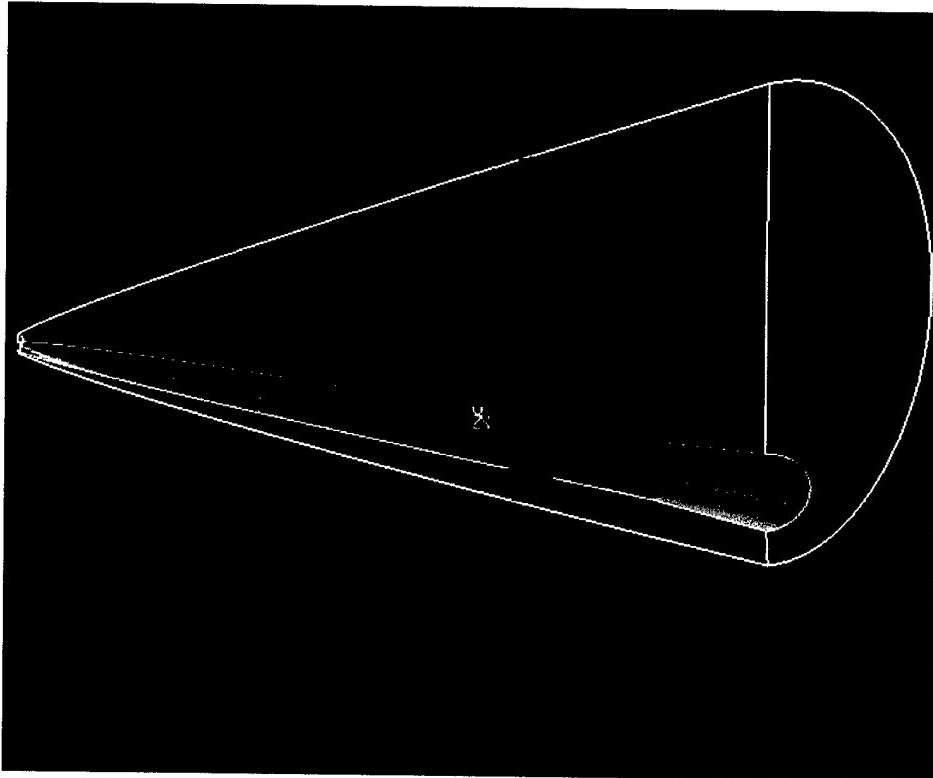
Figure 20. ZEUS Graphic Output (H5 at $\alpha = 15°$ at Mach 6)

bers, it was hypothesized that the combination of high Mach numbers and the density of the grid could not be treated accurately by ZEUS' grid generation algorithm. Although this claim remains unsubstantiated, the following modifications to the grid solved the problem.

Based on a recommendation by Edge [Ref. 29], a coarser grid (24x24) was created and a solution was obtained for the first 20mm of the projectile. At this axial station, the grid was returned to its original dimensions and *convert.x* was run. Pseudo-ZEUS was run for the remaining 520mm showing a properly shaped shock and well ordered grid. The result is depicted in Figure 22.

This problem demonstrates one of the strengths of ZEUS, the ability to stop computation before the end of the projectile and redefine the grid (number of cells and clustering) and zones. Although this feature was used in this research for a special circumstance, its application in the analysis of more complex shapes is obvious.
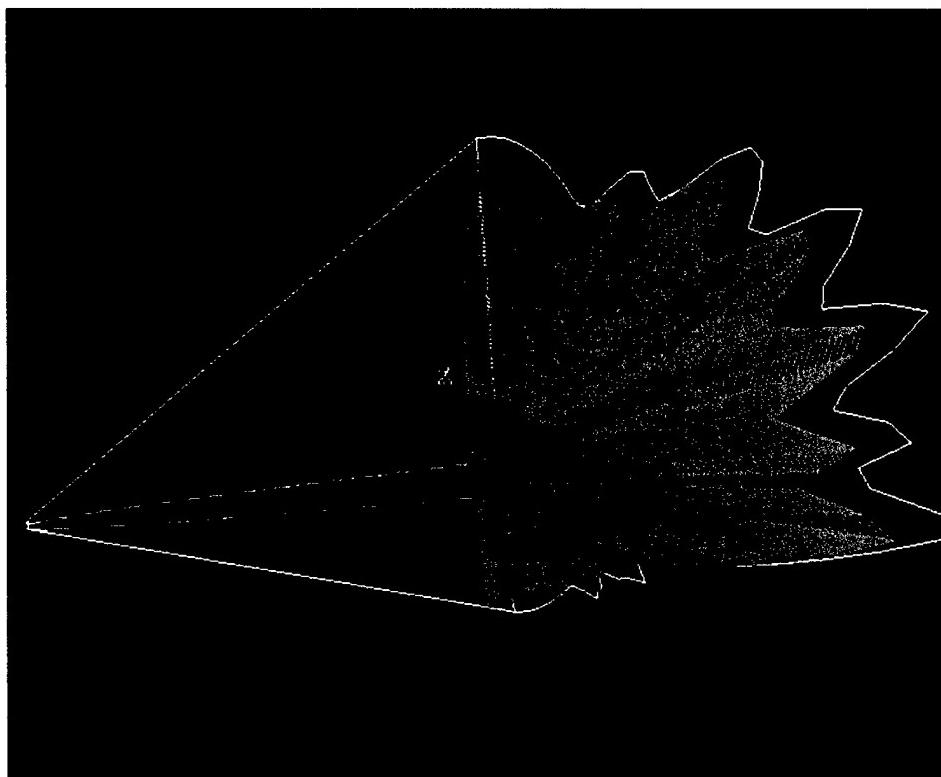
Figure 21. Computational Grid of the H4 at $\alpha = 15^\circ$, Mach 8.2 (Unmodified)
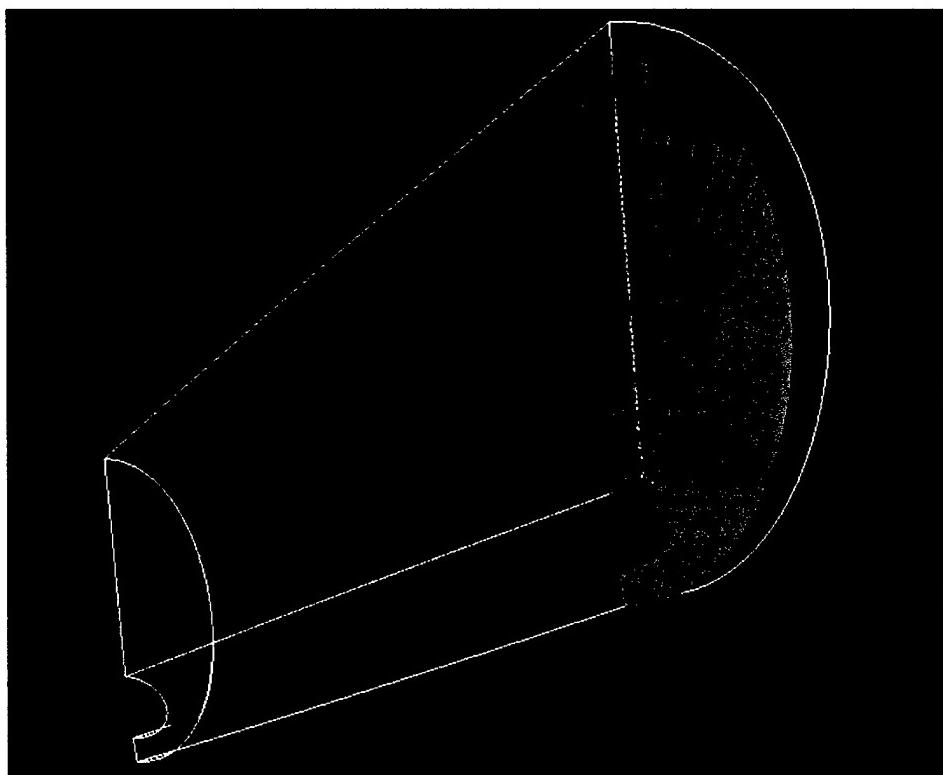
Figure 22. Computational Grid of the H4 at $\alpha = 15°$, Mach 8.2 (Modified)

THIS PAGE INTENTIONALLY LEFT BLANK

VI.   RESULTS

The goal of this chapter is twofold:

- to validate the results computed by ZEUS

- to compare the aerodynamic characteristics of each projectile configuration

In this chapter, lift, drag, moments, center of pressure, normal force distribution, and pressure distribution are analyzed. Particular attention is placed on the effects of the addition of stabilizing surfaces on these characteristics while generalizing the Mach and incidence behavior. A complete listing of the computational results is given in Appendix C.

In this chapter, the non-dimensionalized results are compared. The reference area used was $21,600\text{mm}^2$, the planform area, and the reference length used was 540mm, the projectile length. These are the reference parameters used by Edwards and Roper [Ref. 10] and allowed the results from this study to be compared with previous results.

A.   VALIDATION

Validation of the computational results was conducted using experimental and computed data published by Edwards and Roper [Ref. 10]. In their work, the H3 was tested experimentally at Mach 8.2 and over a wide range of angles of attack. Additionally, runs were conducted using a parabolized Navier-Stokes (PNS) code, CHASM, and Missile DATCOM. For each case, Missile DATCOM was run twice: once using second order shock expansion (SOSE) and once using Newtonian corpuscular impact theory. The results obtained by ZEUS capture the trends demonstrated by the experimental data. For most calculated aerodynamic coefficients, ZEUS tends to under-predict their magnitude. However, the results obtained from ZEUS and the PNS code generally agree.

Figure 23 compares the coefficients of lift. Both CHASM and ZEUS under-predict lift whereas Missile DATCOM over-predicts. As angle of attack is increased, the difference between the ZEUS and CHASM results becomes more evident (the slope of the ZEUS $C_L - \alpha$ curve is smaller than those of other sources). All solvers capture the trend depicted by the experimental data with the absolute error for ZEUS and CHASM being smaller than those for Missile DATCOM.
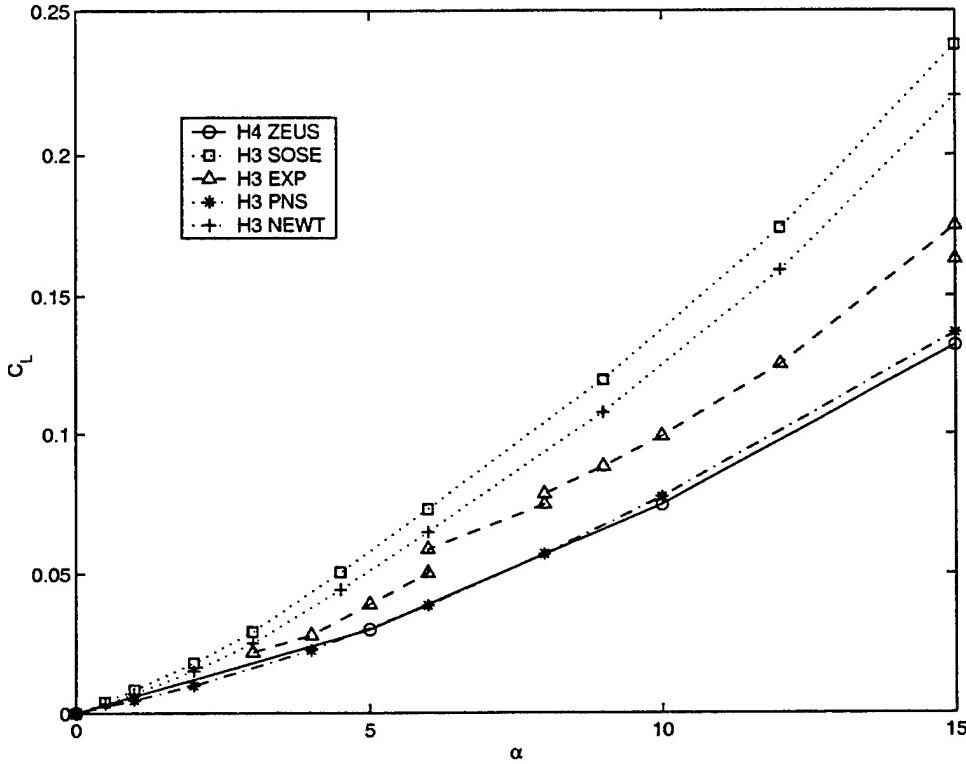
Figure 23. Comparison of C_L versus α Curves

Figure 24 illustrates that ZEUS significantly under-predicts the drag particularly at higher angles of attack. This under-prediction can be partially attributed to ZEUS' inability to account for boundary layer effects. All solvers under-predict the experimental results, however the error incurred by using ZEUS is significantly larger than the other solvers. As with lift, the ZEUS' results are most closely related to the CHASM results. Noteworthy is the fact that the absolute error between ZEUS and CHASM remains relatively constant throughout the range of incidences.

78
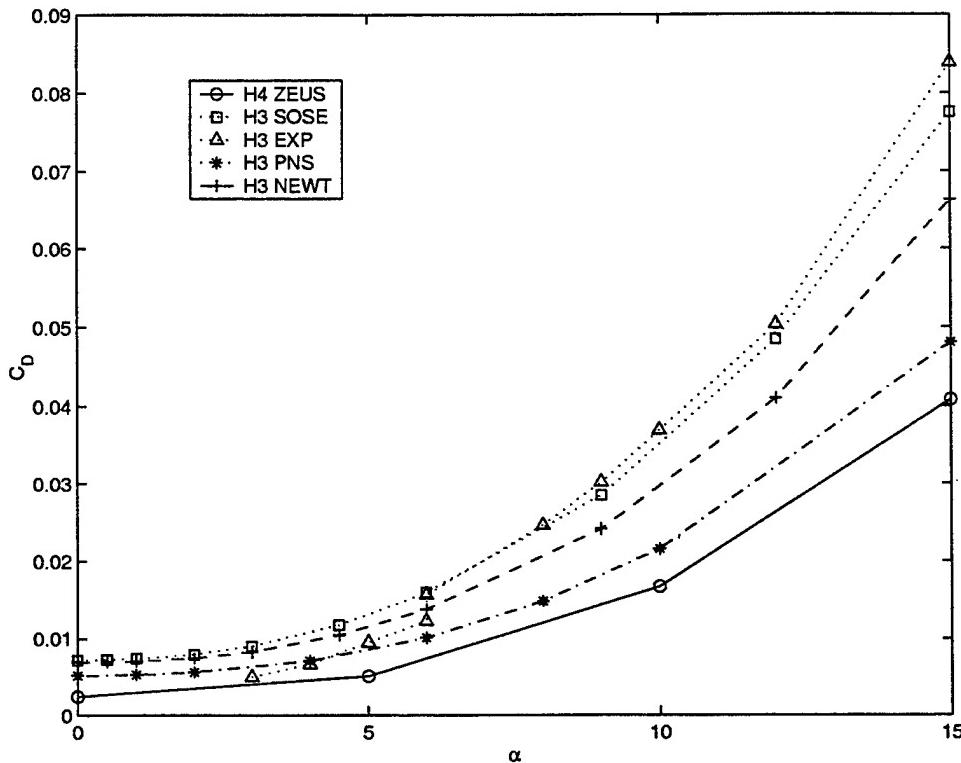
Figure 24. Comparison of C_D versus α Curves

An analysis of the lift-to-drag ratio curve (Figure 25) shows that ZEUS over-predicts the magnitude of the lift-to-drag ratio especially at higher incidences. This error, which is partially based on the under-prediction of the drag, remains relatively constant throughout the range of incidences. More importantly, it demonstrates that the under prediction of drag is more significant than the under prediction of lift. Despite its inability to correctly predict the magnitude of the lift-to-drag ratio, compared to the other solvers, ZEUS was able to best predict the location of $\left(\dfrac{L}{D}\right)_{max}$.

Both CHASM and ZEUS are superior at predicting the coefficient of moment (Figure 26). Results obtained by ZEUS and CHASM are nearly identical for all incidences. Although both tend to under-predict the moment, the results are in better agreement than the Missile DATCOM results which significantly over-predict the moment coefficient.
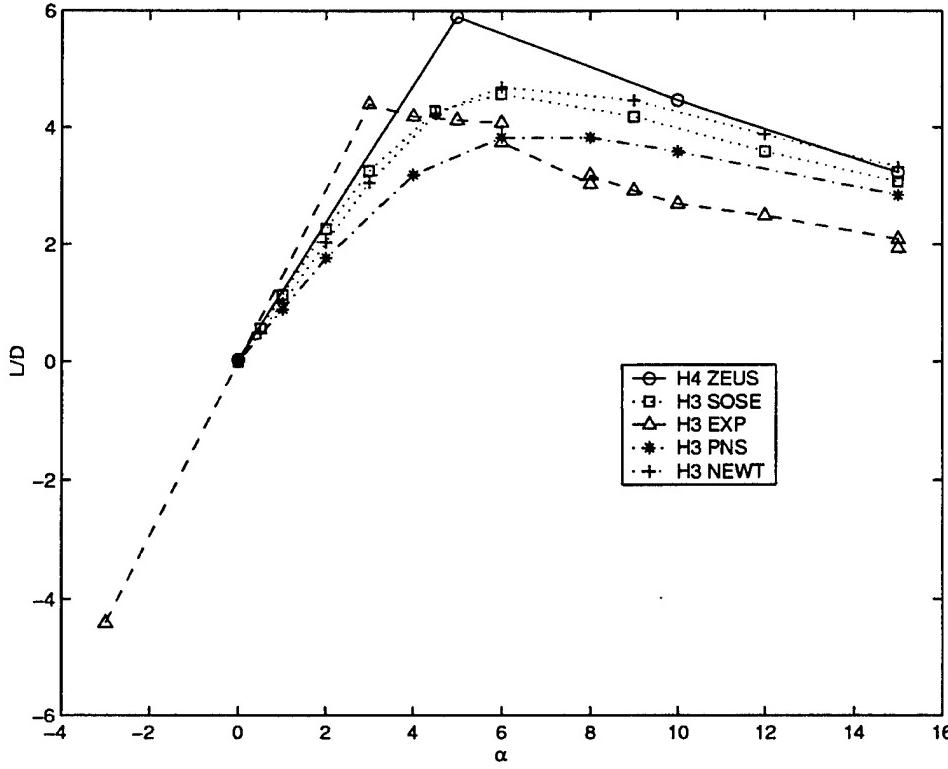
Figure 25. Comparison of $\dfrac{L}{D}$ versus α Curves

B. AERODYNAMIC FORCES AND MOMENTS

Increasing speed slightly decreases lift for all configurations. Graves [Ref. 7], obtaining similar results, ascertained that the reduction in lift was caused by the diminishing effect of the vortex flow. Increases in incidence result in non-linear increases in lift for all projectiles i.e. increasing incidences causes an increase in the slope of the $C_L - \alpha$ curves. This change is the α^2 variation as noted by Kontis et al. [Ref. 9]. Kontis et al. [Ref. 9] also ascertained that the leeside separation of the flow seen in schlieren visualization studies caused increased suction on the leeside thereby accounting for some of the increased lift. Addition of any of the studied stabilizing surfaces results in an increase in lift over the baseline projectile. For all Mach numbers and incidences, the flared projectiles, H5 and H5A, generated the greatest lift. The flared projectiles' ability to achieve greater lift can be attributed

80
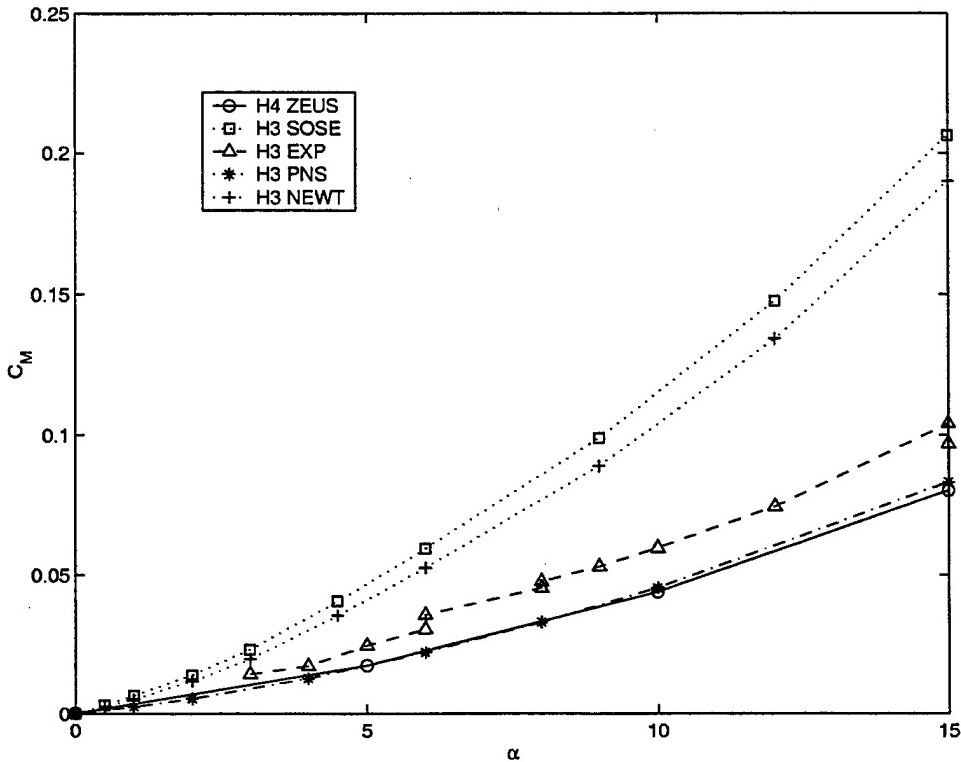
Figure 26. Comparison of C_M versus α Curves

to a larger planform area. Both Graves and Founier et al. [Ref. 7, 8] noted that increased planform area and aspect ratio in elliptical projectiles increased lift. After the flared configurations, the H4A had the next largest lift. A study by Kontis et al. [Ref. 11] shows that the flow interference caused by the addition of the strakes increases lift. This increase in lift can be attributed to an increase in pressure which will be discussed in a later section. The difference in lift between the H5 and H5A was negligible. A comparison of these configurations shows that the addition of the flare appears to dominate. Additionally, because the strakes begin to blend into the flare, they create less cross flow interference than that which is found on the H4A. Figure 27 graphically depicts these results for a sample case (Mach=8.2).

As the Mach number increases, there is a significant decrease in the drag. In supersonic flow, increased speed creates sharper bow shock that are less normal to the body surface. As this occurs, the shock weakens and incurs less drag. The effects

81
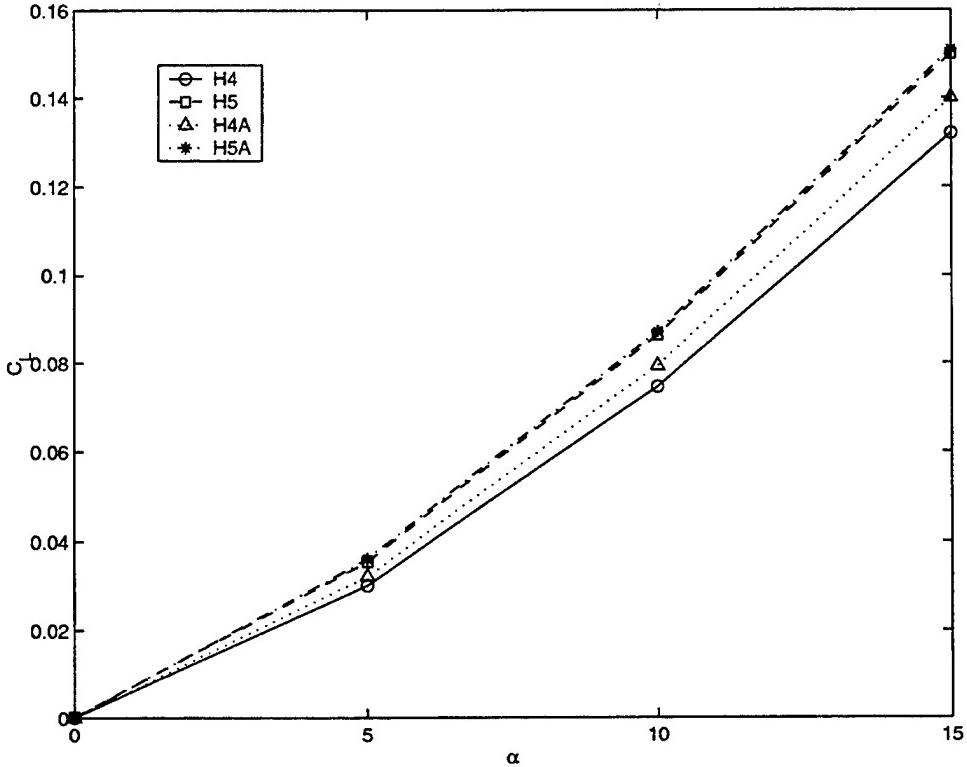
Figure 27. C_L versus α (Mach 8.2)

of increasing incidence on drag are similar to those seen on lift. Like the lift results, increased incidence causes increased drag. Similarly, Figure 28 shows the effects of the addition of control surfaces on drag. A large increase in drag is incurred by the addition of the flare. This finding is consistent with the results computed by Guidos [Ref. 13] on cone-cylinder-flare configurations. Addition of strakes causes a small increase in drag. Larger increases occur at higher incidences based on a greater effect on the bow shock [Ref. 11].

The relationship between lift and drag is depicted in Figure 29. Increasing Mach number results in a significant increase in the lift-to-drag ratio for all configurations particularly at smaller incidences. Generally, $(L/D)_{max}$ occurred between 5° and 10° incidence. $(L/D)_{max}$ was achieved at lower incidences as the Mach number was increased. At non-zero angles of attack, the projectiles with flared tails have significantly lower lift-to-drag ratio indicating that the increase in drag incurred by
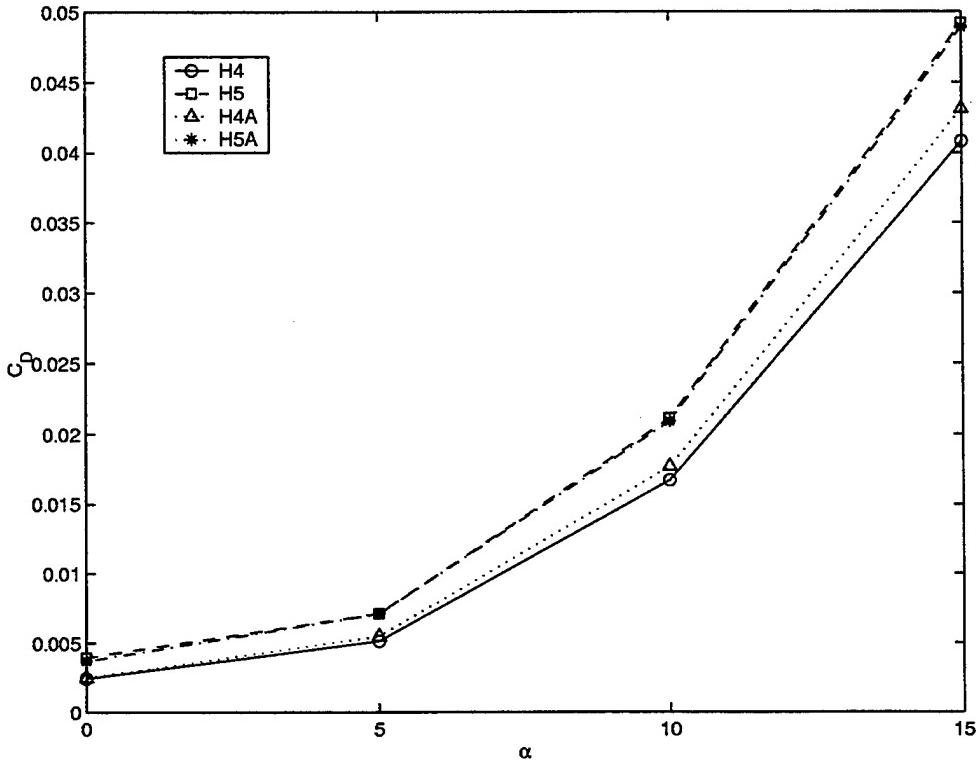
Figure 28. C_D versus α (Mach 8.2)

adding the flare is much greater than the added lift obtained. The addition of strakes on the H4 causes little change in the lift-to-drag ratio. This is inconsistent with the findings of Kontis et al. [Ref. 9] who found that the addition of strakes resulted in a 17% increase in the lift-to-drag ratio. Again, this can be attributed to ZEUS' under prediction of drag.

Generally, the pitching moment decreases with increasing Mach number and increases with increased incidence for all configurations. Figure 30 shows the effects of the stabilizing surfaces. The addition of any studied surface increases the moment. The flare significantly increases the moment, but adding strakes to the projectile with a flare tail has little impact on the moment. The addition of the strakes alone does increase the moment compared to the body alone configuration. However, this increase is not nearly as dramatic as that found by adding a flare. As expected, the configurations demonstrating the highest stability incur the greatest drag.
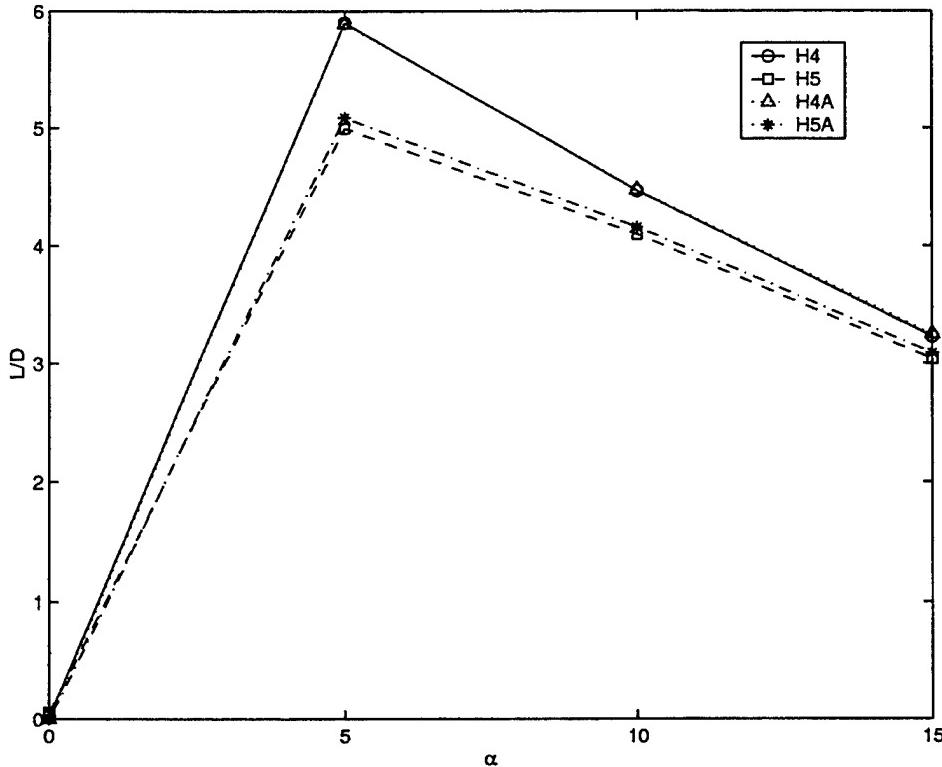
Figure 29. $\dfrac{L}{D}$ versus α (Mach 8.2)

C. PRESSURE DISTRIBUTION AND CENTER OF PRESSURE LOCATION

Surface pressures were calculated for each grid point. The following section analyzes the surface pressures on the windside ($\phi = 0°$) and leeside ($\phi = 180°$) meridians.

For all configurations, increasing angle of attack causes a decrease in the leeside pressure and an increase in the windside pressure. Increasing Mach number has a similar effect. Figures 31, 32, and 33 compare the wind and leeside pressures of the H5, H4A, and H5A, respectively, to those of the H4. These comparisons are over the range of incidences at Mach 8.2.

At all studied angles of attack, there is a significant increase in both the windside and leeside pressures when the flare is added. The pressures acting on the H4 and H5 are identical up to the vicinity of the body-flare junction. The largest

84
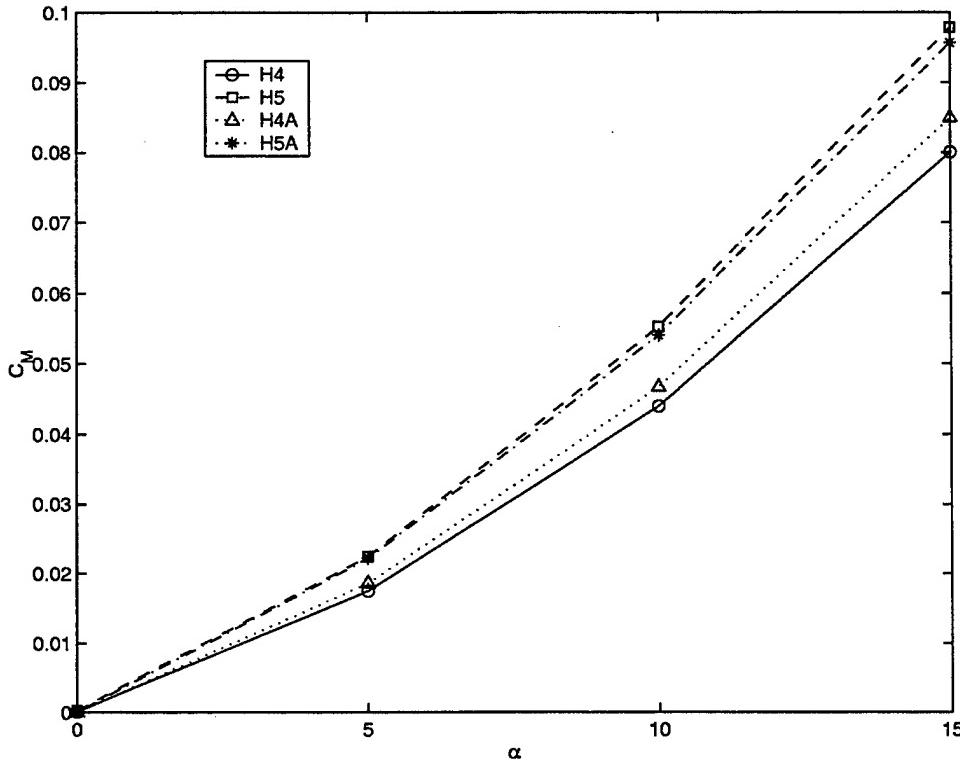
Figure 30. C_M versus α (Mach 8.2)

absolute increases in pressure occur at higher angles of attack. However, in relative terms, the pressure increases at lower angles of attack are larger. It is important to note that Hughson and Chapman [Ref. 12] showed that the magnitude of the pressures just before the flare-body junction are significantly different depending on the nose shape; blunt nosed configurations caused pressures at this location to be dramatically larger.

Application of strakes caused an increase in windside pressures and a decrease in the leeside pressures. These pressure changes are moderate compared to the relatively large increases seen in the flare application. Additionally, they appear to be less dramatic than the pressure differences seen in the application of strakes to a right elliptic cone Kontis et al. [Ref. 9]. As expected, increasing angle of attack results in pressure increases and decreases on the wind and leesides respectively.

Use of both strakes and a flare result in a combination of the previously men-
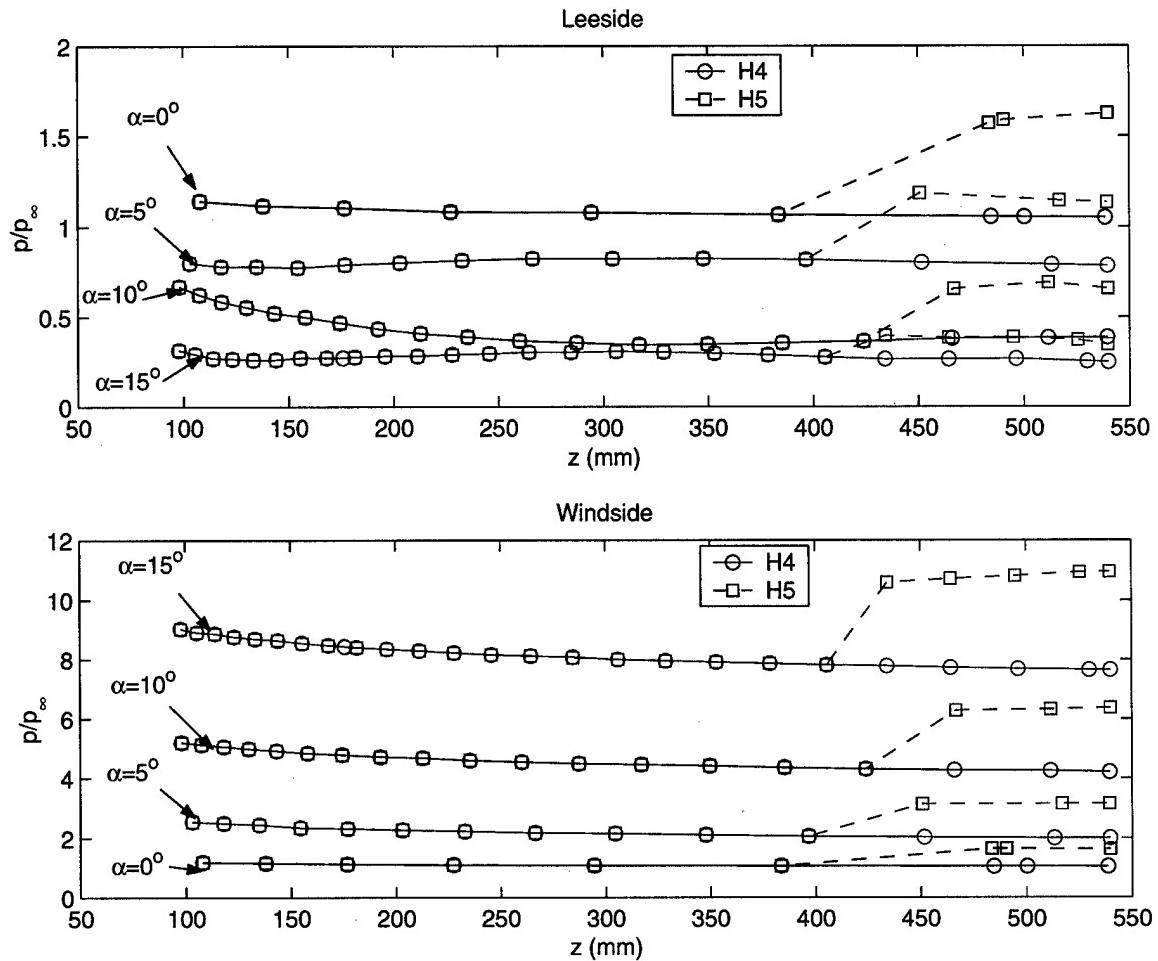
Figure 31. Leeside and Windside Pressure Distributions of the H4 and H5 at Mach 8.2

tioned effects. Again a dramatic pressure increase is seen at the body-flare junction. Increasing angle of attack had the same effect as in the other configurations. Figure 34 compares the pressures on the H5 and H5A to emphasize the effects of the strakes on the flare. Like the effect on the aerodynamic forces and moments, the flare is dominant in the effect of surface pressure. The application of strakes has little effect on the pressure distribution particularly on the windside. The center of pressure was measured as a percentage of the total projectile length. Table IX shows the effects of the control surfaces on the center of pressure. The addition of the flare had the largest impact on the center of pressure resulting in a nearly 5% change. The addition
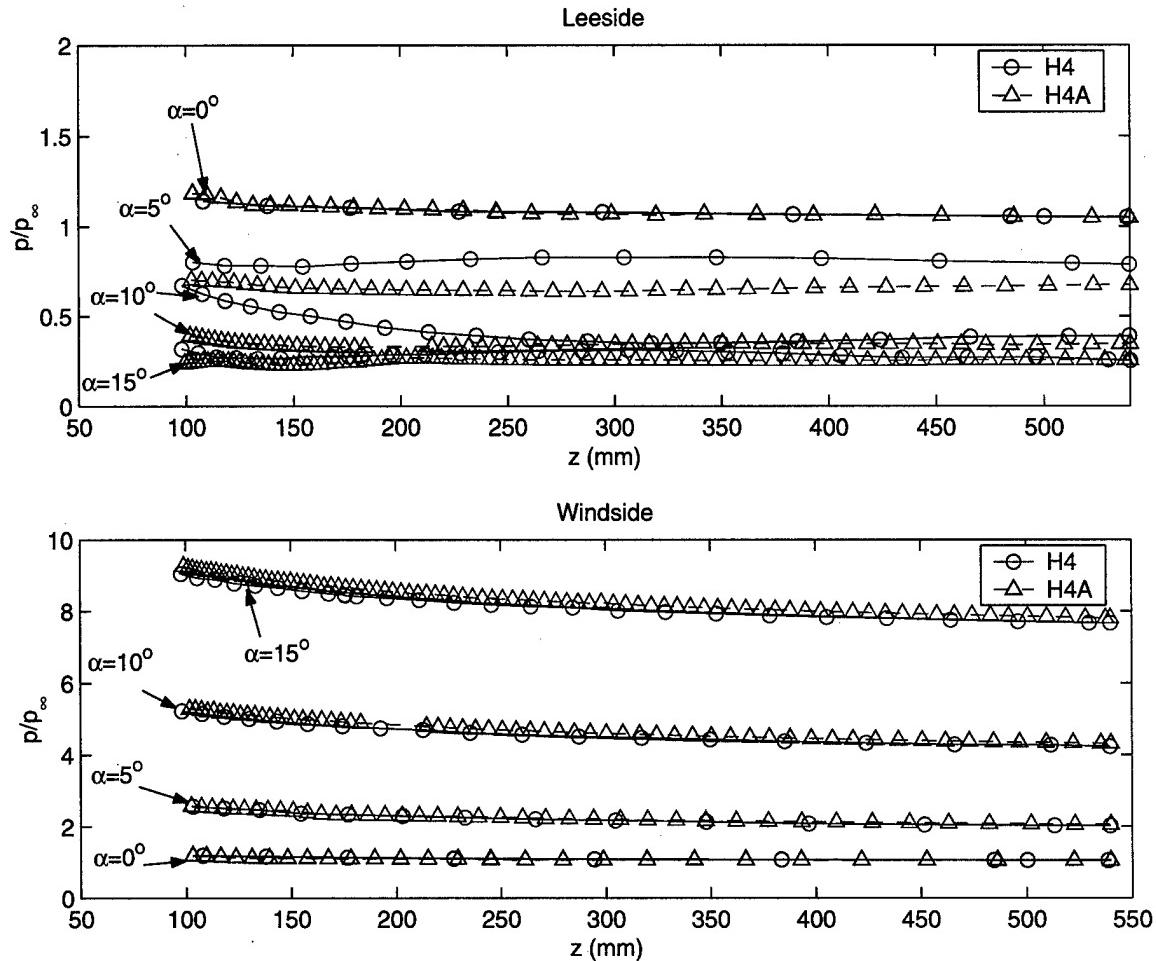
Figure 32. Leeside and Windside Pressure Distributions of the H4 and H4A at Mach 8.2

of strakes had little effect on the center of pressure. These results do not agree with those of Kontis et al. [Ref. 9] who found that the addition of strakes caused a 4% change in the shift (to the rear) in the center of pressure with the addition of strakes.

D. NORMAL FORCE DISTRIBUTION

The following section analyzes the effect of each projectile geometry on the normal force distribution.

Generally, increasing Mach number results in a decrease in the normal force.
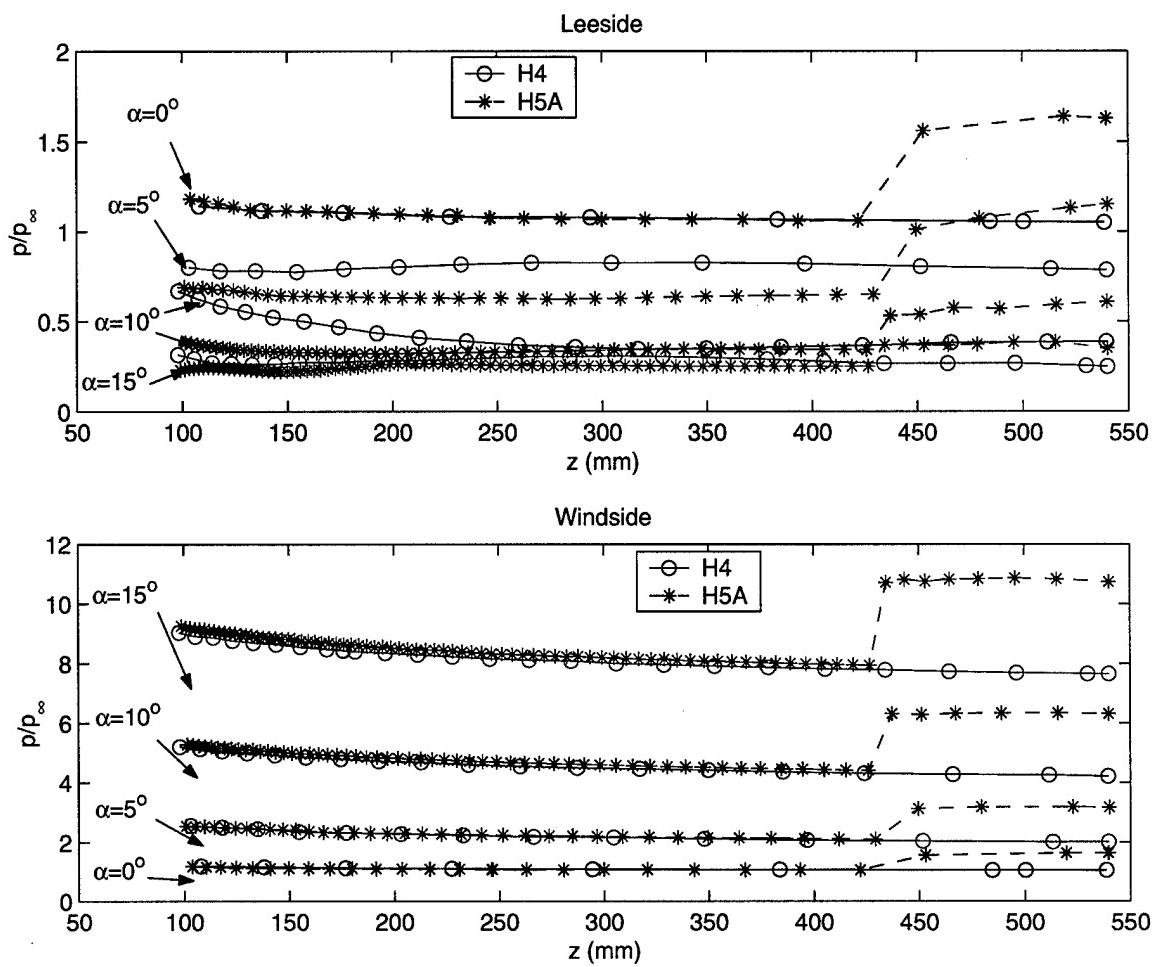
Figure 33. Leeside and Windside Pressure Distributions of the H4 and H5A at Mach 8.2

Projectile	Center of Pressure (% of Projectile Length)
H4	.576
H5	.623
H4A	.575
H5A	.605

Table IX. Centers of Pressure (Mach 8.2)
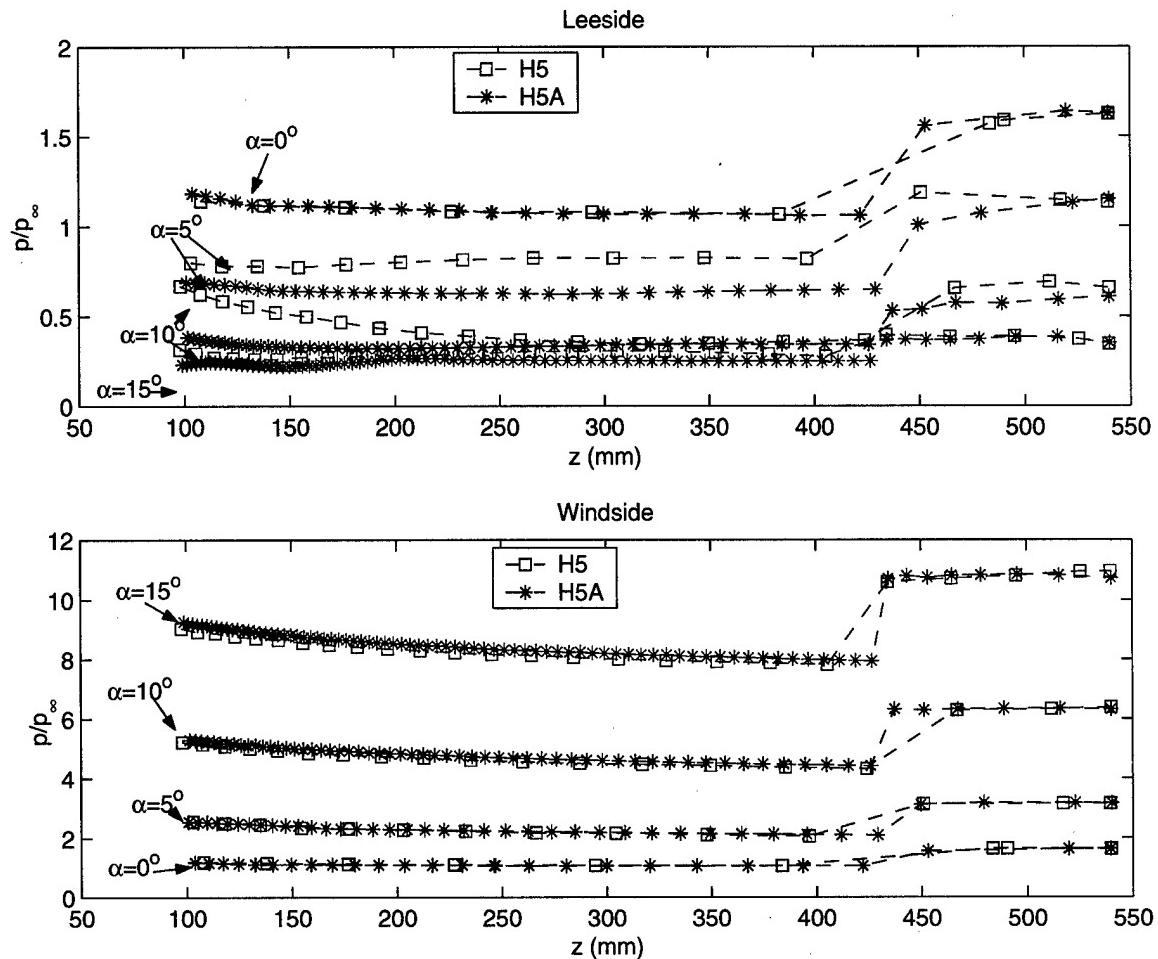
Figure 34. Leeside and Windside Pressure Distributions of the H5 and H5A at Mach 8.2

This decrease in normal force is more dramatic at higher angles of attack. Additionally, higher angles of attack increase the normal force. However, neither speed nor incidence change the normal force distribution trend.

The projectile geometry did have an effect on the normal force distribution. Figure 35 depicts the normal force distribution on each of the projectiles at Mach 8.2 and $\alpha = 5^\circ$. The general shapes of the force distributions are nearly identical for all configuarations for the first 430mm, the location of the body-flare junction. As expected, the straked configurations generated greater normal force, yet the distribution trends were similar to those without strakes.
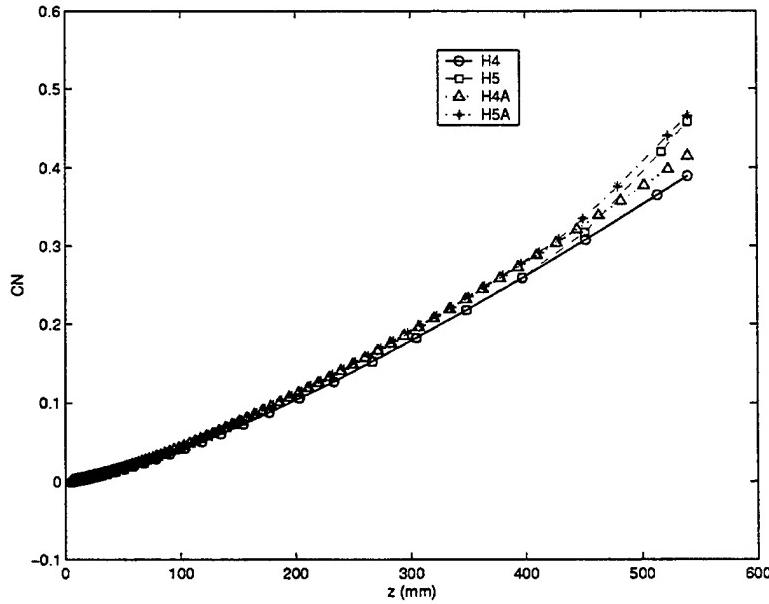
Figure 35. Normal Force Distributions at Mach 8.2, $\alpha = 5°$

At the body-flare junction, a dramatic increase in normal force acting on the projectiles with flares occurs. Figure 36 shows the normal force distribution for $z = 430mm$ to $z = 540mm$. As expected, as the flares taper into the tail of the H5A, the difference in the magnitude of the normal forces become smaller. Also noteworthy is the fact that the normal forces acting on the H4A are larger than those acting on the H5 until $z = 470mm$, over a third of the length of the flare.

E.   PITCHING MOMENT DISTRIBUTION

The pitching moment distribution was analyzed for each projectile configuration. Figure 37 compares the pitching moment distribution for each configuration at Mach 8.2 and $\alpha = 5°$. Like the normal forces, the pitching moments are nearly identical for all configurations from the nose to the vicinity of $z = 250mm$. At this axial station, the effects of the strakes become apparent causing a moderate increase in the moment. Moving aft, the difference between the H4A's and the H4's moment increases demonstrating the influence of increasing strake height on the moment. The moment distribution of the H4 and H5 are identical until the vicinity of the flare-body

90
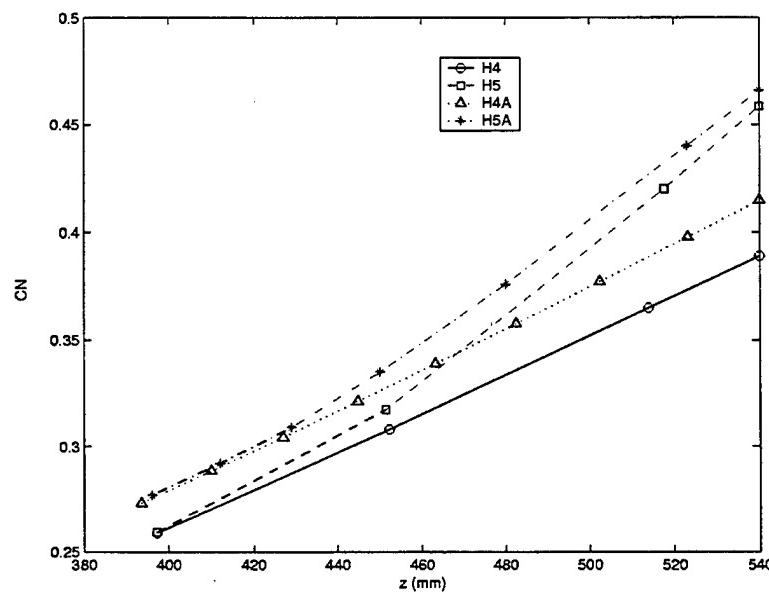
Figure 36. Normal Force Distributions on the Tail Section at Mach 8.2, $\alpha = 5^o$
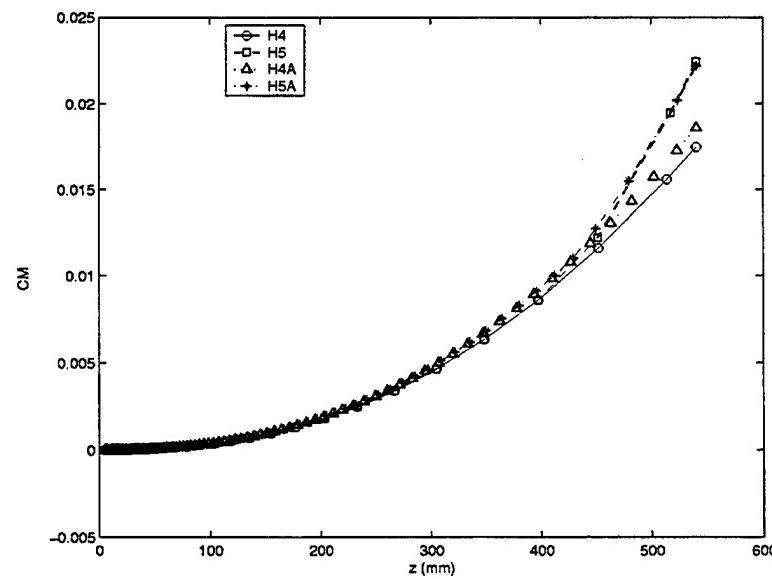
Figure 37. Pitching Moment Distributions at Mach 8.2, $\alpha = 5^o$

junction. Comparing the H4A and H5A shows similar results. At the flare-body junction, the moment of the flare-tailed projectiles increases significantly. The H5A has a greater moment than the H5 until roughly $z = 510$mm when the effects of the strakes become negligible. Similar to the results of the normal force distribution analysis, the pitching moments acting on the H4A are larger than those acting on the H5 until $z = 470$mm.

F.   GRID SENSITIVITY ANALYSIS

As mentioned in Chapter V, problems arose in the grid generation for the configurations with strakes; the grid around the strakes was skewed. Based on the preliminary nature of this study, this grid was used for all runs. However, an improved grid was constructed and tested on one case. This section will describe the results of this analysis.

The construction of the improved grid was more detailed than the single-zoned, clustered mesh used throughout this research. The improved grid was comprised of five zones: one covering each strake and three covering the body surface. The ZEUS grid generation algorithm was unable to properly capture the bow shock shape so the mesh was made uniform and more coarse (35x35).

Figure 38 shows the improved grid around the strake on the H4A. A test run was conducted on the H4A at Mach 8.2, $\alpha = 5°$. The results are shown in Table X. The analysis shows that the improved grid had almost no impact on the calculation of the axial force and a small effect on the normal force. The improved grid resulted in a roughly 2% increase in the normal force. Of the coefficients calculated by ZEUS, the one most affected by the improved grid was the pitching moment coefficient. Use of the new mesh increased the pitching moment coefficient over 2%. The center of pressure shifted slightly toward the base with the improved grid. Overall, the improved grid had a small impact on the results.

A second grid sensitivity analysis was conducted to attempt to account for the
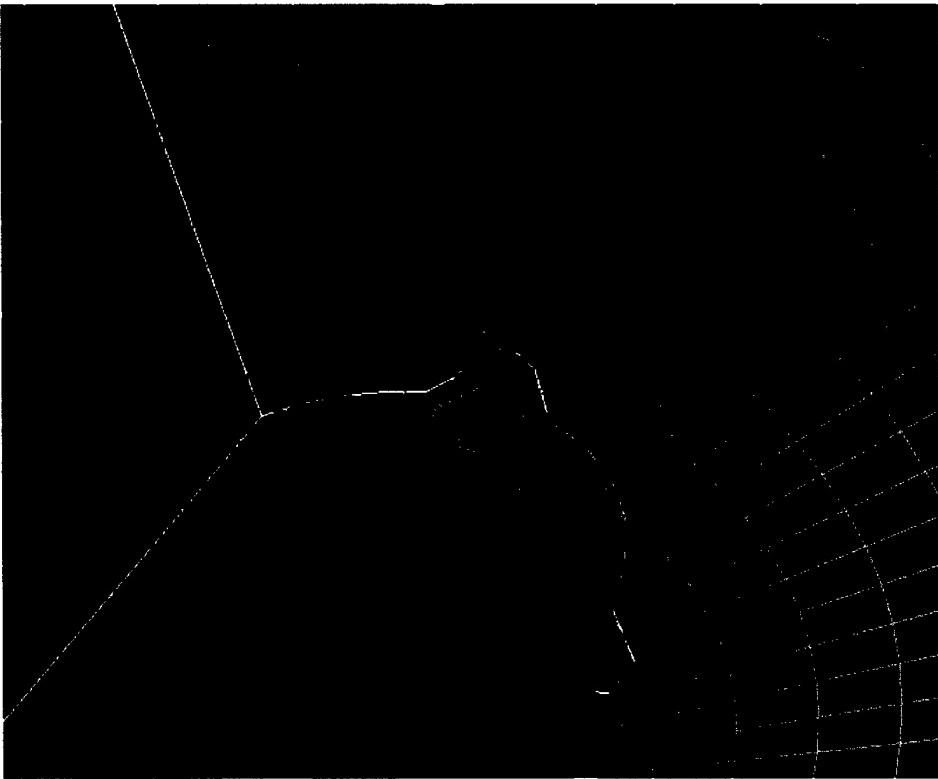
Figure 38. Improved Grid for the H4A

under prediction of drag. Because the mesh is "stretched" on the leeside at higher angles of attack, the mesh was clustered normally. The analysis was conducted on the H4 at all studied angles of attack resulting in negligible differences with the previously computed results.

	C_X	C_Z	C_M	XCP
Original	3.2571E-02	2.6372E-03	1.8578E-02	5.8942E+00
Improved	3.3298E-02	2.6288E-03	1.9095E-02	5.9668E+00
Error	2.18%	-0.32%	2.71%	1.22%

Table X. Grid Sensitivity Results

93

THIS PAGE INTENTIONALLY LEFT BLANK

VII.   CONCLUSIONS

Four projectile configurations with elliptical cross-sections have been analyzed using a spatial marching Euler solver. The results were compared to available experimental and computed data. The evaluation of the results shows:

- ZEUS is able to reasonably predict trends in aerodynamic characteristics for these projectiles. Neglected boundary layer effects and difficulty constructing the computational grid about the strakes cause error often leading to poor agreement with experimental results. ZEUS is capable of predicting lift and moment to a level of accuracy usually found in the more computationally intensive PNS codes.

- Generally, the effects caused by changes in Mach number and moderate changes in angle of attack are similar for all studied configurations.

- The application of strakes, a flare, or both cause an increase in lift. The current flare design induces greater lift than the strakes.

- Addition of a flare significantly increases drag. Although the strakes also generate additional drag, the amount of drag created by the flare is much greater.

- For all configurations, the maximum lift-to-drag ratio occurs at approximately 5°. The drag penalty incurred by increasing lift is largest for the projectiles with flares. The difference in lift-to-drag ratio for designs with and without strakes is very small.

- The configurations with the flare had the highest pitching moment coefficient. Adding strakes increases the coefficient of pitching moment regardless of whether a flare is present.

- Addition of strakes causes increased pressure on the windside and a decrease in pressure on the leeside. Application of a tail flare results in a dramatic increase in pressure on both the wind and lee sides from the flare-body junction to the base.

- Normal force distribution trends are identical for all projectiles until the body-flare junction. The flare causes a large increase in the force distribution. This increase is more dramatic for the flared projectile without strakes.

95

- The pitching moment for the flared configurations increases significantly starting at the body-flare junction. As the strakes become more prominent on the projectiles (toward the tail of the H4A or the body-flare junction of the H5A), the pitching moment increases.

APPENDIX A. CODES USED FOR THE CONSTRUCTION OF THE PROJECTILE GEOMETRY FILES

The following are MATLAB codes designed to create files describing the geometry of each projectile. The output of each code is a file in PLOT3D format that can be used in CM3DT and ZEUS. The codes shown create half of the projectile (to be used for incidence computations only). Additionally, the nose generation codes have been omitted. But for several modifications the nose generation codes are nearly identical to the projectile generation codes. As noted in Chapter III, the following modifications were made to full projectile codes:

- The projectile length is reduced to 10mm

- The number of points defining to nose are reduced in the circumferential direction and increased in the axial direction

- The results are output to coincide with the CM3DT coordinate system

1. THE H4 GENERATOR–*H4GENERATOR.M*

```
%h4generator.m
%This code creates a .P3D file for an elliptical projectile.
%3MAY2000

data=[];X=[];Y=[];Z=[];

z=linspace(0,540,109);
theta=linspace(pi/2,-pi/2,50);
for j=1:50
  for i=1:109;
    a=sqrt(15.*z(i))/3;
    b=sqrt(15.*z(i))/5;
    y=a*cos(theta(j));
    x=b*sin(theta(j));
    X=[X,x];Y=[Y,y];Z=[Z,z(i)];
  end
```

```
end

one=1;
jmax=j;
imax=109;
data=[X,Y,Z];

fid = fopen('h4.grd','w');
fprintf(fid,'%d \n',one);
fprintf(fid,'%d %d %d \n',imax,jmax,one);
fprintf(fid,'%12.8e %12.8e  %12.8e \n',data);
fclose(fid);
```

2.   THE H4A GENERATOR–*H4AGENERATOR.M*

```
%h4agenerator.m
%This code creates a .P3D file for an elliptical projectile.
%3MAY2000

data=[];X1=[];Y1=[];Z1=[];X=[];Y=[];Z=[];imax=0;jmax=0;
z=linspace(0,540,109);

%I. Creates body shell from bottom(pi/2) to the bottom, right strake
theta=linspace(pi/2,.9614,10);
for i=1:10
      a=sqrt(15.*z)/3;
      b=sqrt(15.*z)/5;
      y=a*cos(theta(i));
      x=b*sin(theta(i));
      X=[X,x];Y=[Y,y];Z=[Z,z];
end
imax=imax+i;

%II.  Creates bottom,right strake
%A.  Creates side from surface to top of strake (straight line).
%This is accomplished by determining the equation of the line at each
%value of z.  This line is broken into 4 intervals (size h) and
%the point for (y=y1+ih,x=my+b) is determined for all z.
phi=.8844; % This the angle to the strake top point.
a1=sqrt(2.268.*z);
b1=sqrt(.9833.*z);
y1=a1.*cos(phi);
```

98

```
x1=b1.*sin(phi);

for i=1:3
  for j=1:109
    if j==1%this conditional statement eliminates singularity at
          %origin
      x=0;y=0;
      X=[X,x];Y=[Y,y];Z=[Z,z(1)];
    else
      m=(x1(j)-X(j+981))/(y1(j)-Y(j+981));
      bint=x1(j)-m.*y1(j);
      h=abs(y1(j)-Y(j+981))/4;
      y=Y(j+981)+(h*i);
      x=m*y+bint;
      X=[X,x];Y=[Y,y];Z=[Z,z(j)];
    end
  end
end
imax=imax+i;

%B.  This loop will construct the top of the strake.
phi=linspace(phi, .7158,5);
for i=1:5
  y=a1.*cos(phi(i));
  x=b1*sin(phi(i));
  X=[X,x];Y=[Y,y];Z=[Z,z];
end
imax=imax+i;

%C.  This loop will construct the other side of the strake.  This is
%an modeling approximation of the actual shape (straight).  The shape
%is modeled with a circular arc varying arc length.  The circle from
%which this is derived has a varying center and radius that are
%calculated in the code.
eta=.5203;%angle to dummy (modeled) point on body surface
zeta=.6578;%angle to actual point on body surface
y2=a.*cos(eta);
x2=b.*sin(eta);
y3=a.*cos(zeta);
x3=b.*sin(zeta);

for i=1:10
```

```
  for j=1:109
    if j==1%this conditional statement eliminates singularity at
          %origin
      X=[X,0];Y=[Y,0];Z=[Z,0];
    else
      %Approximate three points of circle
      ymidbad=(y2(j)-Y(j+1853))/2+Y(j+1853);
      xmidbad=(x2(j)-X(j+1853))/2+X(j+1853);
      ymidba=(y3(j)-Y(j+1853))/2+Y(j+1853);
      xmidba=(x3(j)-X(j+1853))/2+X(j+1853);
      ythird=(ymidbad-ymidba)/2+ymidba;
      xthird=(xmidbad-xmidba)/2+xmidba;
      %Calculate center and radius of circle
      x4=X(j+1853);
      y4=Y(j+1853);
      h=(((xthird^2-x2(j)^2)+(ythird^2-y2(j)^2))/...
        (2*(xthird-x2(j)))-((x2(j)^2-x4^2)+...
        (y2(j)^2-y4^2))/(2*(x2(j)-x4)))/((y4-y2(j))/...
        (x2(j)-x4)+(ythird-y2(j))/(xthird-x2(j)));
      k=((x2(j)^2-x4^2)+(y2(j)^2-y4^2))/...
        (2*(x2(j)-x4))+((y4-y2(j))/(x2(j)-x4))*h;
      r=sqrt((x2(j)-k)^2+(y2(j)-h)^2);
      deltax=abs(x2(j)-x4)/10;
      x=x4+deltax*i;
      y=-sqrt(r^2-(x-k)^2)+h;
      X=[X,x];Y=[Y,y];Z=[Z,z(j)];
    end
  end
end
imax=imax+i;

%III. Creates projectile shell between strakes
theta=linspace(eta,-eta,20);
for i=1:20
  y=a*cos(theta(i));
  x=b*sin(theta(i));
  X=[X,x];Y=[Y,y];Z=[Z,z];
end
imax=imax+i;

%IV.  Creates top strake.  Since this strake is a reflection of
%the previously modeled strake reflected about the y-axis.
```

```
for i=1:28
  j=3052-(i*109)+1;
  X=[X,-X(j:j+108)];Y=[Y,Y(j:j+108)];Z=[Z,Z(j:j+108)];
end
imax=imax+i;

one=1;
data=[X,Y,Z];
jmax=109;
fid = fopen('h4a1.grd','w');
fprintf(fid,'%d \n',one);
fprintf(fid,'%d %d %d \n',jmax,imax,one);
fprintf(fid,'%12.8e %12.8e  %12.8e \n',data);
fclose(fid);
```

3.   THE H5 GENERATOR–*H5GENERATOR.M*

```
%h5generator.m
%This code creates the H5 elliptical projectile with flared tail.
%03MAY2000

data=[];X=[];Y=[];Z=[];
theta=linspace(pi/2,-pi/2,50);
bigx=35*cos(theta);
bigy=22.85*sin(theta);

for j=1:50
  for z=0:5:430
    a=sqrt(15.*z)/3;
    b=sqrt(15.*z)/5;
    x=a*cos(theta(j));
    y=b*sin(theta(j));
    X=[X,x];Y=[Y,y];Z=[Z,z];
  end

  dxdz=(bigx(j)-x)/(540-430);
  dydz=(bigy(j)-y)/(540-430);
  bxz=x-dxdz*430;
  byz=y-dydz*430;

  for i=435:5:540
    x=dxdz*i+bxz;
```

```
      y=dydz*i+byz;
      z=i;
      X=[X,x];Y=[Y,y];Z=[Z,z];
   end
end

one=1;
jmax=j;
imax=109;
data=[Y,X,Z];

fid = fopen('h5.grd','w');
fprintf(fid,'%d \n',one);
fprintf(fid,'%d %d %d \n',imax,jmax,one);
fprintf(fid,'%12.8e %12.8e  %12.8e \n',data);
fclose(fid);
```

4.   THE H5A GENERATOR–*H5AGENERATOR.M*

```
%h5agenerator.m
%This code creates the H5 elliptical projectile with flared tail.
%03MAY2000

data=[];X=[];Y=[];Z=[];imax=0;jmax=0;x2=[];y2=[];x3=[];y3=[];

theta=linspace(pi/2,.9614,10);
bigy=35*cos(theta);
bigx=22.85*sin(theta);

for i=1:10
  for z=0:5:430
    a=sqrt(15.*z)/3;
    b=sqrt(15.*z)/5;
    y=a*cos(theta(i));
    x=b*sin(theta(i));
    X=[X,x];Y=[Y,y];Z=[Z,z];
  end

  dxdz=(bigx(i)-x)/(540-430);
  dydz=(bigy(i)-y)/(540-430);
  bxz=x-dxdz*430;
  byz=y-dydz*430;
```

```
    for j=435:5:540
       x=dxdz*j+bxz;
       y=dydz*j+byz;
       z=j;
       X=[X,x];Y=[Y,y];Z=[Z,z];
    end
end
imax=imax+i;

z=linspace(0,540,109);
phi=.8844; % This the angle to the strake top point.
a1=sqrt(2.268.*z);
b1=sqrt(.9833.*z);
y1=a1.*cos(phi);
x1=b1.*sin(phi);

for i=1:3
   for j=1:109
      if j==1%this conditional statement eliminates singularity at
             %origin
        x=0;y=0;
        X=[X,x];Y=[Y,y];Z=[Z,z(1)];
      else
        m=(x1(j)-X(j+981))/(y1(j)-Y(j+981));
      bint=x1(j)-m.*y1(j);
      h=abs(y1(j)-Y(j+981))/4;
        y=Y(j+981)+(h*i);
        x=m*y+bint;
        X=[X,x];Y=[Y,y];Z=[Z,z(j)];
      end
   end
end
imax=imax+i;

%B.  This loop will construct the top of the strake.
phi=linspace(phi, .7158,5);
for i=1:5
   y=a1.*cos(phi(i));
   x=b1*sin(phi(i));
   X=[X,x];Y=[Y,y];Z=[Z,z];
end
```

```
imax=imax+i;

%C.  This loop will construct the other side of the strake.  This
%is an modeling approximation of the actual shape (straight).  The
%shape is modeled with a circular arc varying arc length.  The
%circle from which this is derived has a varying center and radius
%that are calculated in the code.
eta=.5203;%angle to dummy (modeled) point on body surface
zeta=.6578;%angle to actual point on body surface

for z=0:5:430
  a=sqrt(15*z)/3;
  b=sqrt(15*z)/5;
  y=a*cos(eta);
  x=b*sin(eta);
  yy=a*cos(zeta);
  xx=b*sin(zeta);
  x2=[x2,x];y2=[y2,y];x3=[x3,xx];y3=[y3,yy];
end

bigy=35*cos(eta);
bigx=22.85*sin(eta);
bigyy=35*cos(zeta);
bigxx=22.85*sin(zeta);
dxdz=(bigx-x)/(540-430);
dydz=(bigy-y)/(540-430);
bxz=x-dxdz*430;
byz=y-dydz*430;
dxxdz=(bigxx-xx)/(540-430);
dyydz=(bigyy-yy)/(540-430);
bxxz=xx-dxxdz*430;
byyz=yy-dyydz*430;

for j=435:5:540
  x=dxdz*j+bxz;
  y=dydz*j+byz;
  xx=dxxdz*j+bxxz;
  yy=dyydz*j+byyz;
  x2=[x2,x];y2=[y2,y];x3=[x3,xx];y3=[y3,yy];
end

z=linspace(0,540,109);
```

```
for i=1:10
  for j=1:109
      if j==1%this conditional statement eliminates singularity
            %at origin   ·
        X=[X,0];Y=[Y,0];Z=[Z,0];
      else
%Approximate three points of circle
        ymidbad=(y2(j)-Y(j+1853))/2+Y(j+1853);
        xmidbad=(x2(j)-X(j+1853))/2+X(j+1853);
        ymidba=(y3(j)-Y(j+1853))/2+Y(j+1853);
        xmidba=(x3(j)-X(j+1853))/2+X(j+1853);
        ythird=(ymidbad-ymidba)/2+ymidba;
        xthird=(xmidbad-xmidba)/2+xmidba;
        %Calculate center and radius of circle
        x4=X(j+1853);
        y4=Y(j+1853);
        h=(((xthird^2-x2(j)^2)+(ythird^2-y2(j)^2))/...
          (2*(xthird-x2(j)))-((x2(j)^2-x4^2)+(y2(j)^2-y4^2))/...
          (2*(x2(j)-x4)))/((y4-y2(j))/(x2(j)-x4)+(ythird-...
          y2(j))/(xthird-x2(j)));
       k=((x2(j)^2-x4^2)+(y2(j)^2-y4^2))/(2*(x2(j)-x4))+...
          ((y4-y2(j))/(x2(j)-x4))*h;
        r=sqrt((x2(j)-k)^2+(y2(j)-h)^2);
        deltax=(x2(j)-x4)/10;
        x=x4+deltax*i;
y=-sqrt(r^2-(x-k)^2)+h;
        if j<109
  X=[X,x];Y=[Y,y];Z=[Z,z(j)];
else
  theta=asin(x/22.85);
  X=[X,x];Y=[Y,35*cos(theta)];Z=[Z,z(j)];
end
     end
  end
end
imax=imax+i;

%III. Creates projectile shell between strakes
theta=linspace(eta,0,10);
bigy=35*cos(theta);
bigx=22.85*sin(theta);
```

```
for i=1:10
  for z=0:5:430
    a=sqrt(15.*z)/3;
    b=sqrt(15.*z)/5;
    y=a*cos(theta(i));
    x=b*sin(theta(i));
    X=[X,x];Y=[Y,y];Z=[Z,z];
  end

    dxdz=(bigx(i)-x)/(540-430);
    dydz=(bigy(i)-y)/(540-430);
    bxz=x-dxdz*430;
    byz=y-dydz*430;

  for j=435:5:540
    x=dxdz*j+bxz;
    y=dydz*j+byz;
    z=j;
    X=[X,x]; Y=[Y,y];Z=[Z,z];
  end
end
imax=imax+i;


%IV.  Creates top strake.  Since this strake is a reflection
%of the previously modeled strake reflected about the y-axis.
for i=1:37
  k=4034-(i*109);
  X=[X,-X(k:k+108)];Y=[Y,Y(k:k+108)];Z=[Z,Z(k:k+108)];
end
for p=1:109
  Y(p)=0;
  Y(p+8064)=0;
end
imax=imax+i;

one=1;
jmax=109;
data=[X,Y,Z];

fid = fopen('h5a.grd','w');
fprintf(fid,'%d \n',one);
fprintf(fid,'%d %d %d \n',jmax,imax,one);
```

```
fprintf(fid,'%12.8e %12.8e  %12.8e \n',data);
fclose(fid);
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. SAMPLE CM3DT AND ZEUS INPUT FILES (H4 $\alpha = 0^O$ MACH 5)

1. CM3DT INPUT FILES

a. Geometry Pre-Processor Input File–*preproc.inp*

```
$input
  isa=1,lmax=9,fname='h4anose.inp',
$end
```

b. CM3DT Solver Input File–*CM3DT.INP*

```
CM3DT
 $INPUT
   AMINF=5,
   ILAM=1,
   NMAX=11,MMAX=18,LMAX=9,
   KMAX=1000,KOUT=1000,
   NSEG=0,XB0=0.0,
   kense(4)=1,
   rn=.5,xbb=9*8,alpha=0.0,
   ktape=1000,kense(5)=1
 $END
 $ZEUS
   na=18,ma=24,zstart=5.00,z0=0.0,ibl=1,
   x0=0.0
 $END
```

2. ZEUS INPUT FILES

a. Initial Conditions Input File–*initc.inp*

```
0.0 ALPHA
0.0 BETA
5.0 XMINF
1.0 PINF
0.00001 DINF
```

b. Integration Control Input File–*intcntrl.inp*

```
1 IMOD
5.00 ZSTART
540.00 ZETAEND
1500 KEND
```

```
0.90 FCFL
1.00 XKI
1 IAPR
0.90 DFAC
0 IVIS
```

c.   Zone Dimensions Input File–*zondim.inp*

```
1 IZN
72 NA
72 MA
72 MAZ(IZN)
0 INPANG
ANGMAZ(IZN)
```

d.   Boundary Conditions Input File–*bc.inp*

```
1 IASYM
0 NXKE
0 NSUR
1 ISHOCK
50.00 1.25 540.75 RSHOCK PHI1SH PHI2SH
```

e.   Output Control Input File–*outcntrl.inp*

```
1 IEFORCE
1696.46 AEREF
1 XLREF
1 IPLT3D
20 INCP3D
1 IOZEUS
500 IPRINT
18 NSKIP
18 MSKIP
50 ISKIP
0 JSPPR
50 KSKIP
100 NPRT
100 DELZA
1000 IPLOT
200 DELZC
0 IPLOTN
```

APPENDIX C. COMPUTED RESULTS FROM ZEUS

H4					
Mach 5					
α	C_X	C_L	C_Z	C_D	C_M	X_{CP}
0	9.60171E-06	9.60171E-06	5.31056E-03	5.31056E-03	2.43380E-05	1.36877E+03
5	3.14319E-02	3.08883E-02	4.86514E-03	7.58610E-03	1.70176E-02	2.92363E+02
10	8.97250E-02	8.74438E-02	5.28713E-03	2.07874E-02	5.11541E-02	3.07865E+02
15	1.57475E-01	1.50349E-01	6.80190E-03	4.73277E-02	9.11934E-02	3.12715E+02
Mach 6					
0	4.03579E-05	4.03579E-05	3.91452E-03	3.91452E-03	4.59032E-05	6.14199E+02
5	3.23951E-02	3.19527E-02	3.66156E-03	6.47105E-03	0.018046296	3.00817E+02
10	8.48458E-02	8.27962E-02	4.38042E-03	1.90472E-02	0.048651492	3.09642E+02
15	1.49157E-01	1.42506E-01	6.05958E-03	4.44578E-02	0.086538066	3.13297E+02
Mach 7					
0	6.02926E-05	6.02926E-05	3.07053E-03	3.07053E-03	7.07231E-05	6.33418E+02
5	3.18681E-02	3.14881E-02	2.96838E-03	5.73457E-03	0.018026492	3.05457E+02
10	8.04736E-02	7.85803E-02	3.86242E-03	1.77778E-02	0.046272805	3.10502E+02
15	1.42990E-01	1.36660E-01	5.63259E-03	4.24492E-02	0.083029835	3.13562E+02
Mach 8.2					
0	6.91120E-05	6.91120E-05	5.26036E+01	2.43535E-03	9.6953E-05	7.57533E+02
5	3.05624E-02	3.02309E-02	5.33295E+01	5.12325E-03	0.017462706	3.08545E+02
10	7.63588E-02	7.45924E-02	7.54232E+01	1.66983E-02	0.043980967	3.11028E+02
15	1.37767E-01	1.31696E-01	1.14896E+02	4.07946E-02	0.080021005	3.13655E+02

H5					
Mach 5					
α	C_X	C_L	C_Z	C_D	C_M	X_{CP}
0	7.80023E-05	7.80023E-05	8.45546E-03	8.45546E-03	8.25882E-05	5.71749E+02
5	3.53338E-02	3.45212E-02	7.78014E-03	1.08301E-02	2.05997E-02	3.14823E+02
10	9.99963E-02	9.70130E-02	8.43176E-03	2.56678E-02	6.05754E-02	3.27119E+02
15	1.74542E-01	1.65774E-01	1.08967E-02	5.57002E-02	1.06854E-01	3.30584E+02
Mach 6					
0	1.39519E-04	1.39519E-04	6.28116E-03	6.28116E-03	1.31353E-04	5.08393E+02
5	3.68539E-02	3.62023E-02	5.86782E-03	9.05753E-03	2.21267E-02	3.24210E+02
10	9.58653E-02	9.31848E-02	7.04940E-03	2.35891E-02	5.87419E-02	3.30888E+02
15	1.67111E-01	1.58873E-01	9.82764E-03	5.27443E-02	1.02990E-01	3.32799E+02
Mach 7					
0	1.90365E-04	1.90365E-04	4.95074E-03	4.95074E-03	1.84084E-04	5.22185E+02
5	3.68175E-02	3.62621E-02	4.76458E-03	7.95531E-03	2.25495E-02	3.30731E+02
10	9.21259E-02	8.96366E-02	6.27528E-03	2.21774E-02	5.69305E-02	3.33700E+02
15	1.61716E-01	1.53814E-01	9.23907E-03	5.07794E-02	1.00173E-01	3.34499E+02
Mach 8.2					
0	2.35248E-04	2.35248E-04	3.93597E-03	3.93597E-03	2.43470E-04	5.58872E+02
5	3.60126E-02	3.55291E-02	3.97547E-03	7.09905E-03	2.24378E-02	3.36448E+02
10	8.86731E-02	8.63288E-02	5.74273E-03	2.10534E-02	5.52365E-02	3.36377E+02
15	1.57282E-01	1.49637E-01	8.83181E-03	4.92384E-02	9.78781E-02	3.36046E+02

H4A					
Mach 5					
α	C_X	C_L	C_Z	C_D	C_M	X_{CP}
0	1.33857E-05	1.33857E-05	5.63023E-03	5.63023E-03	1.63766E-07	6.60657E+00
5	3.61795E-02	3.55745E-02	5.36176E-03	8.49461E-03	1.97623E-02	2.94963E+02
10	9.45856E-02	9.21578E-02	5.70616E-03	2.20441E-02	5.38735E-02	3.07570E+02
15	1.65864E-01	1.58380E-01	7.08005E-03	4.97677E-02	9.57605E-02	3.11766E+02
Mach 6					
0	6.43222E-06	6.43222E-06	4.11024E-03	4.11024E-03	3.31669E-07	2.78444E+01
5	3.55308E-02	3.50472E-02	3.99755E-03	7.07906E-03	1.98006E-02	3.00931E+02
10	8.97375E-02	8.75606E-02	4.68514E-03	2.01967E-02	5.14367E-02	3.09523E+02
15	1.57426E-01	1.50452E-01	6.22171E-03	4.67547E-02	9.10322E-02	3.12257E+02
Mach 7					
0	3.10100E-06	3.10100E-06	3.19044E-03	3.19044E-03	9.52992E-08	1.65951E+01
5	3.43067E-02	3.38963E-02	3.21061E-03	6.18841E-03	1.93819E-02	3.05078E+02
10	8.53019E-02	8.32954E-02	4.09174E-03	1.88421E-02	4.89908E-02	3.10135E+02
15	1.51519E-01	1.44880E-01	5.70671E-03	4.47284E-02	8.77040E-02	3.12568E+02
Mach 8.2					
0	1.42867E-06	1.42867E-06	2.49925E-03	2.49925E-03	1.54847E-07	5.85283E+01
5	3.25705E-02	3.22167E-02	2.63716E-03	5.46583E-03	1.85784E-02	3.08019E+02
10	8.12259E-02	7.93569E-02	3.65685E-03	1.77060E-02	4.67130E-02	3.10554E+02
15	1.46669E-01	1.40293E-01	5.32412E-03	4.31034E-02	8.49345E-02	3.12708E+02

			H5A			
			Mach 5			
α	C_X	C_L	C_Z	C_D	C_M	X_{CP}
0	1.53205E-05	1.53205E-05	8.19352E-03	8.19352E-03	1.74144E-06	6.13806E+01
5	3.93450E-02	3.85136E-02	7.82106E-03	1.12204E-02	2.24772E-02	3.08493E+02
10	1.01158E-01	9.81789E-02	8.30778E-03	1.81834E-02	5.97174E-02	3.18782E+02
15	1.75503E-01	1.66824E-01	1.04264E-02	5.54946E-02	1.04414E-01	3.21271E+02
			Mach 6			
0	7.20787E-06	7.20787E-06	6.00361E-03	6.00361E-03	1.14270E-06	8.56085E+01
5	3.89064E-02	3.82469E-02	5.86806E-03	9.23664E-03	2.27715E-02	3.16056E+02
10	9.68843E-02	9.42146E-02	6.89759E-03	1.63611E-02	5.78242E-02	3.22292E+02
15	1.67941E-01	1.59819E-01	9.26991E-03	5.24203E-02	1.00491E-01	3.23121E+02
			Mach 7			
0	4.64259E-06	4.64259E-06	4.68972E-03	4.68972E-03	1.48262E-06	1.72450E+02
5	3.79903E-02	3.74329E-02	4.73671E-03	8.02976E-03	2.26586E-02	3.22073E+02
10	9.29519E-02	9.04836E-02	6.08162E-03	1.51636E-02	5.58558E-02	3.24492E+02
15	1.62732E-01	1.54963E-01	8.59352E-03	5.04188E-02	9.78095E-02	3.24564E+02
			Mach 8.2			
0	2.76933E-06	2.76933E-06	3.69011E-03	3.69011E-03	1.59348E-06	3.10717E+02
5	3.65974E-02	3.61170E-02	3.91411E-03	7.08889E-03	2.21796E-02	3.27263E+02
10	8.93852E-02	8.70725E-02	5.49806E-03	1.42332E-02	5.40592E-02	3.26587E+02
15	1.58575E-01	1.51073E-01	8.10824E-03	4.88742E-02	9.56867E-02	3.25844E+02

LIST OF REFERENCES

[1] Hogg, I. *Tank Killing*, Sarpedon, New York, 1996.

[2] Moore, F.G. *Engineering Codes for Aeropredictions: State-of-the-Art and New Methods.* Advisory Group for Aerospeace Research and Development (APGARD) Report 804, Special Course on Missile Aerodynamics, June 1994.

[3] Hall, D.W., Hines, R.W., Baltakis, F.P., and Wardlaw, A.B. *Coupled Inviscid/Viscous Aerodynamic Predictions for Supersonic Tactical Missiles*, AIAA Paper 90-0617, January 1990.

[4] Hughson, M.C. and Cottrell, C.J. *Hypersonic Prediction Comparisons with Experimental Data for a Cone-Cylinder at Mach 6.86*, AIAA Paper 89-0524, January 1989.

[5] Edge, H. and Clark, J. *Graphical User Interface for ZEUS*, ARL-TR-1093, Army Research Laboratory, June 1996.

[6] Nielsen, J.M. *Missile Aerodynamics*, McGraw-Hill Book Company, Inc., New York, 1960.

[7] Graves, E.B. *Aerodynamic Characteristics of a Monoplanar Missile Concept with Bodies of Circular and Elliptical Cross Sections*, NASA TM-74079, Langley Research Center, December 1977.

[8] Fournier, R.H., Spencer, B., and Corlett, W.A. *Supersonic Aerodynamic Characteristics of a Series of Related Bodies with Cross-Sectional Ellipticity*, NASA TN D-3539, Langley Research Center, August 1966.

[9] Kontis, K., Stollery, J.L., and Edwards, J.A. *Hypersonic Effectiveness of Slender Lifting Elliptic Cones with and without Strakes*, AIAA Paper 97-0521, January 1997.

[10] Edwards, J.A. and Roper, J.J. *A Computational Assessment of Static and Dynamic Coefficients for the H3 Hypervelocity Projectile*, AIAA Paper 97-0640, January 1997.

[11] Kontis, K., Stollery, J.L., and Edwards, J.A. Hypersonic Effectiveness of a Lifting Cone with and without Strakes. *Journal of Spacecraft and Rockets*, 37(1):21-27, January-February 2000.

[12] Hughson, M.C. and Chapman, G.T. *A Numerical Study of Flare Stabilized Projectiles at Mach 7.0*, AIAA Paper 87-0446, January 1987.

[13] Guidos, B.J. *Static Aerodynamic CFD Analysis for 120-mm Hypersonic KE Projectile Design*, ARL-MR-184, Army Research Laboratory, September 1994.

[14] Danberg, J.E., Sigal, A., and Celmins, I. *Prediction and Comparison with Measurements of the Aerodynamic Characteristics of Flare-Stabalized XM910 Prototypes*, BRL-MR-3752, Ballistic Research Laboratory, May 1989.

[15] Priolo, F.J. and Wardlaw, A.B. *Supersonic Non-Circular Missile Computations*, AIAA Paper 88-0278, January 1988.

[16] Hall, D.W. *Inviscid Aerodynamic Predictions for Ballistic Reentry Vehicles with Ablated Nosetips*, PH.D. Dissertation, The University of Pennsylvania, 1979.

[17] Hall, D.W. *Calculation of Inviscid Supersonic Flow Over Ablated Nosetips*, AIAA Paper 79-0342, January 1979.

[18] Moretti, G. *Computation of Shock Layers About Ablated, Blunt-Nosed Bodies*, POLY-M/AE Report No. 77-14, Polytechnic Institute of New York, August, 1977.

[19] Hall, D.W. Application of $\lambda-$Differencing Schemes to Hypersonic Inviscid Flows. *Computers and Fluids*, 17:113-126, 1989.

[20] Moretti, G. The $\lambda-$Scheme. *Computational Fluids*, 7: 191-205, 1979.

[21] MacCormack, R.W. *The Effect of Viscosity in Hypersonic Impact Cratering*, AIAA Paper 69-354, May 1969.

[22] Wardlaw, A.B. and Davis, S.F. *A Second Order Godunov Method for Supersonic Tactical Missiles*, NSWC TR 86-506, Naval Surface Warfare Center, December 1986.

[23] Warlaw, A.B., Solomon, J.M, and Baltakis, F.P. Euler Space Marching Methods Applied to Missiles. *Tactical Missile Aerodynamics*, 104: 669-722, 1986.

[24] Wardlaw, A.B., Baltakis, F.P., Priolo, F.J., and Solomon, J.M. Space Marching Euler Solvers. *Tactical Missile Aerodynamics: Prediction Methodology*, 142: 379-444, 1992.

[25] Hall, D.W. and Dougherty, C.M. *Inviscid Aerodynamic Predictions for Ballistic Reentry Vehicles with Ablated Nosetips: User's Manual*, Technical Report to U.S. Air Force F04701-77-C-0126, September 1979.

[26] Hall, D.W. *The CM3DT Nosetip Code (ZEUS-Coupled Version)*, Technical Report to U.S. Army DAAL01-98-P-0799, October 1998.

[27] Wardlaw, A.B. and Davis, S.F. *Applying the ZEUS Code*, NSWC TR 86-508, Naval Surface Warfare Center, December 1986.

[28] Davis, S.F. Simplified Second-Order Godunov-Type Methods. *SIAM Journal of Scientific and Statistical Computing*, 9 (3): 445-473, May 1988.

[29] Edge, H., U.S. Army Research Laboratory, AMSRL-WM-BC, Electronic Mail Message to the Author, March 21, 2000.

[30] Edge, H. U.S. Army Research Laboratory, AMSRL-WM-BC, Interviews with the Author, November, 1999.

[31] Farrar, C.L. and Leeming, D.W. *Military Ballistics–A Basic Manual*, Pergamon Press, Inc., New York, 1983.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center................................2
   8725 John J. Kingman Road., Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library................................2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Beny Neta................................1
   Naval Postgraduate School
   Department of Mathematics
   Code MA/Nd
   Monterey, CA 93943-5101

4. Peter Plostins................................1
   U.S. Army Research Laboratory
   ATTN: AMSRL-WM-BC
   Aberdeen Proving Ground, MD 21005-5066

5. Harris Edge................................1
   U.S. Army Research Laboratory
   ATTN: AMSRL-WM-BC
   Aberdeen Proving Ground, MD 21005-5066

6. John A. Edwards................................1
   Defense Research Agency
   Fort Halstead
   Sevenoaks
   TN14 7BP
   UK

7. Michael Morgan, Chairman...1
   Naval Postgraduate School
   Department of Mathematics
   Code MA/Mw
   Monterey, CA 93943-5101

8. COL David C. Arney, Chairman...1
   Department of Mathematics
   United States Military Academy
   West Point, NY 10996

9. CPT Christopher Boyle...4
   Department of Mathematics
   United States Military Academy
   West Point, NY 10996